Claim Charts for ’480 and ’766 Patents

US 10,992,480 B2 – Date of Patent: April 27, 2021

’480 Patent Claim 1		
Feature	**Support from Patent**	**Chase Bank’s Use**
Method for performing an electronic transaction between a first transaction party and a second transaction party	P.10, Col 1 lines 57-67, Col 2 lines 1-3: At least this object is achieved in the present invention by a method for performing an electronic transaction between a first transaction party and a second transaction party using an electronic device operated by the first transaction party. The method comprises providing authentication data in a memory of said electronic device, which authentication data are inaccessible to a user of the electronic device; providing authentication software in said electronic device, the authentication data being accessible to said authentication software; activating the authentication software to generate a digital signature from the authentication data; providing the digital signature to the second transaction party.	EXHIBIT 14: The Chase cardholder is the first transaction party and the second transaction party is the merchant that the Chase cardholder is purchasing from with Chase Pay + Samsung Pay. EXHIBIT 9, p. 5-10: Chase directs Chase cardholder-users to add Chase cardholder data to Samsung Pay for mobile use in stores, when shopping online and in apps. . EXHIBIT 9, p. 13: Many Chase Visa and Mastercard credit cards, debit and Chase Liquid cards work with Samsung Pay.
using an electronic device operated by the first transaction party,	*See* above: a method for performing an electronic transaction between a first transaction party and a second transaction party using an electronic device operated by the first transaction party	EXHIBIT 9: the first transaction party, a Chase cardholder uses a Samsung mobile device
the electronic device having an operating system creating a run-time environment for user applications	P. 13, Col 8 lines 33-34: The operating system 48 creates a run-time environment for any user applications.	EXHIBIT 1, p. 5: In this section, we examine the basic structure of **the Android OS,** which KNOX is built on. Recall that a hardware processor provides two modes, user and privileged. Operating systems use both of these modes for various functions. The portion running in privileged mode is **called the kernel**. OS kernels are among the most rigorously engineered pieces of software in the world, because they must perform many functions, all with the power of the processor’s privileged mode. For example, any time data arrives for the phone from the Internet, the OS kernel first chooses whether to even allow the data to proceed, or to drop it if it seems unwanted. If the data is allowed, **the kernel examines it and decides which application on the phone the data is intended for.** The kernel

		then places the data in the app's memory, and notifies the app that data has arrived. If the app then wishes to send a reply, the app's reply is sent by repeating this whole process in reverse.
the method comprising: providing a private key in a memory of said electronic device, said memory being a secure part of a Basic In Out System or any other secure location in said electronic device	Abstract: lines 2-5: The method provides authentication data and authentication software to an electronic device and preferably stored in a secure storage location or other location inaccessible to the user or the operating system of the device. P. 12, Col 6 lines 45-53: The installation in a new device is completed according to cell 20. The encoded authentication table is stored in a secure part of the BIOS, where it may only be retrieved by the BIOS and not by the operating system. This secure part of the BIOS **may also be any other secure location in the device**. For instance, it may be a separate part of a hard drive of a computer, which part may not be accessible to the operating system, but only to the BIOS **and/or authentication software**. P. 13, Col 8 lines 62-65: In or behind the console 52, there is a secure area 62 only accessible to the BIOS. The secure area 62 comprises applications and storage locations, which are not reported to the operating system 48.	EXHIBIT 1, p. 6: KNOX maintains signing keys are only accessible in the TrustZone Secure World. EXHIBIT 3, p.7/EXHIBIT 4, p.7: TIMA Client Certificate Management (CCM) - **Enables cryptographic keys to be sequestered in a secure area of the device**, so that private key information is never exposed to the Android operating system. EXHIBIT 10, p.2: Tokenization is the process of replacing essential credit card credentials — the 16-digit primary account number (PAN), for instance — with a substitute value. Called a token PAN or digitized PAN (DPAN), the token protects the real card number from theft and misuse. Payment tokenization adds a cryptogram to the mix. **The cryptogram contains unique authentication data generated by the smartphone device.** When making a purchase, the user's device sends the payment token, along with a cryptogram, to the merchant POS, which relays them to the payment network for approval by the issuer (Chase). **The cryptogram is generated using a cryptographic key based on the algorithm furnished by the card network.** **This key, stored in the device's trusted execution environment (TEE), called TrustZone**, is static or dynamic, depending on the card brand/network. The card network holds the master key to their card product and use it to generate a **unique derived key (UDK) for each cardholder, which remains unchanged for the lifetime of the card.**

		EXHIBIT 9, p. 13: For Chase card holders using a mobile wallet, the number on the Chase card "is replaced with a secure device account number (also called a token). This token is passed onto the merchant for payment instead of your actual card number." EXHIBIT 11, p. 1: **All Samsung Pay** tokens and **keys are stored within the TEE** in encrypted form using a hardware-based device key that is unique to each device.
which private key is inaccessible to a user of said electronic device	Abstract: lines 2-5: The method provides authentication data and authentication software to an electronic device and preferably stored in a secure storage location or other location inaccessible to the user or the operating system of the device. P. 12, Col 6 lines 45-53: The installation in a new device is completed according to cell 20. The encoded authentication table is stored in a secure part of the BIOS, where it may only be retrieved by the BIOS and not by the operating system. This secure part of the BIOS may also be any other secure location in the device. For instance, it may be a separate part of a hard drive of a computer, which part may not be accessible to the operating system, but only to the BIOS **and/or authentication software**. P. 13, Col 8 lines 62-65: In or behind the console 52, there is a secure area 62 only accessible to the BIOS. The secure area 62 comprises applications and storage locations, which are not reported to the operating system 48.	EXHIBIT 1, p.15: The TIMA KeyStore stores all application keys in the TrustZone Secure World storage EXHIBIT 1, p.20: The Secure World is a hardware-isolated environment in which highly sensitive software executes. (...) Normal world software can never access the data used by Secure World software. EXHIBIT 4, p.7: TIMA KeyStore Samsung KNOX builds and improves upon the basic Android KeyStore foundation to provide enterprise class mobile application and data security **with a hardware-based TIMA KeyStore. KNOX uses a TrustZone-based TIMA KeyStore for cryptographic key storage.** EXHIBIT 11, p. 1: All Samsung Pay tokens and keys are **stored within the TEE in encrypted form using a hardware-based device key that is unique to each device.**

wherein the private key is encrypted when the private key is stored in said memory,	P.13, Col 7 lines 57-65: Next, according to cell 36, the authentication software verifies the authentication table and requests BIOS-specific data from the BIOS to encrypt the authentication table again. According to cell 38 the BIOS sends BIOS-specific data, for example any common encryption key, to the authentication software in reply. The method is completed according to cell 40, wherein the authentication table is encrypted and stored in a secure part of the BIOS. P. 8, lines 28-29: The authentication software preferably has its own unique serial number, software ID **and/or private encryption key**.	EXHIBIT 1, p.32: Keys stored in the TIMA KeyStore are further encrypted with the device-unique hardware key (DUHK), and can only be decrypted from within TrustZone Secure World on the same device. EXHIBIT 2, p.41: Keys stored in the TIMA KeyStore are further encrypted with the *device-unique hardware key* (DUHK) and can only be decrypted from within TrustZone Secure World on the same device. EXHIBIT 11, p. 1: **All Samsung Pay** tokens and **keys are stored within the TEE in encrypted form using a hardware-based device key that is unique to each device.**
a decryption key for decrypting the private key being incorporated in the electronic device	P.13, Col 7 lines 57-65: Next, according to cell 36, the authentication software verifies the authentication table and requests BIOS-specific data from the BIOS to encrypt the authentication table again. According to cell 38 the BIOS sends BIOS-specific data, for example any common encryption key, to the authentication software in reply.	EXHIBIT 1, p.19: Device-Unique Hardware Key (DUHK). The DUHK is a device-unique symmetric key that is set in hardware at manufacture time in the Samsung factory. EXHIBIT 2, p. 25: The DUHK is a device-unique symmetric key that is set in hardware at manufacture time in the Samsung factory….. The DUHK is only accessible to a hardware cryptography module and is not directly exposed to any software. However, software can request for data to be encrypted

		and decrypted by the DUHK. Data encrypted by the DUHK becomes bound to the device, because it cannot be decrypted on any other device. **The DUHK is typically used to encrypt other cryptographic keys.**
said decryption key being inaccessible to said user, to any user-operated software	P 8, lines 28-29: The authentication software preferably has its own unique serial number, software ID and/or private encryption key. P. 13, Col 8 lines 29-32: The BIOS 44 may comprise an encryption key 46. Such an encryption key 46 may be used to encrypt data and only another party who knows the encryption key 46 may be able to decrypt the data. P. 15, Col 11 lines 56-61: One encryption layer may be decrypted by the authentication software 54. The other encryption layer may be an encryption/decryption application stored in the electronic device using a device specific encryption key. Such an application may be stored in the BIOS or in a secure area 58 (as indicated in Fig. 3) and may use the encryption key 46.	EXHIBIT 1, p.19 The DUHK is only accessible to a hardware cryptography module and **is not directly exposed to any software**
and to said operating system of said electronic device,	P. 13, Col. 8, lines 62-67: In or behind the console 52, there is a secure area 62 only accessible to the BIOS. The secure area 62 comprises applications and storage locations, which are not reported to the operating system 48. **That means that the operating system 48 does not know that the applications and data in the storage locations are present** P. 14, Col 9., lines 7-11: An authentication table may be securely stored in the secure storage location 60. In such a part of the computer, commonly seen as a part of the BIOS 44, the authentication table is **unreachable for the operating system 48 and thus for a user.**	EXHIBIT 1, p.20: The Secure World is a hardware-isolated environment in which highly sensitive software executes. (...)

wherein said memory is inaccessible to said operating system of said electronic device, thereby rendering the authentication data inaccessible to said user;	P. 13, Col 8 lines 62-65: In or behind the console 52, there is a secure area 62 only accessible to the BIOS. The secure area 62 comprises applications and storage locations, which are not reported to the operating system 48. P. 14, Col 9 lines 7-11: An authentication table may be securely stored in the secure storage location 60. In such a part of the computer, *commonly seen* as a part of the BIOS 44, the authentication table is unreachable for the operating system 48 and **thus for a user**.	EXHIBIT 1, p.20: Most of the system as we know it, including the kernel and middleware, as well as all apps, execute in what is called the normal world. Normal world software can never access the data used by Secure World software. EXHIBIT 4, p.7: TIMA KeyStore Samsung KNOX builds and improves upon the basic Android KeyStore foundation to provide enterprise class mobile application and data security **with a hardware-based TIMA KeyStore. KNOX uses a TrustZone-based TIMA KeyStore for cryptographic key storage.**
wherein the private key is decrypted in a secure processing environment inaccessible to said user and to any user operated software	P. 11, Col 4 lines 66-67, P. 12 Col 5 lines 1-3: The authentication software preferably runs in a separate operating environment in the BIOS **or in a console and is independent from and inaccessible to the operating system (OS) on the device.** P. 14, Col 9 lines 42 – 51: To prevent that the authentication data becomes accessible to a user, the authentication data should only be decrypted in a secure processing environment such as a 'Ring Zero' processing environment, which is embedded in a processing unit of commonly used personal computers. Such a secure processing environment may only be accessible to selected software applications, preferably not to user-operated software applications. Further, a decryption algorithm **and/or a decryption key should not be obtainable to any user**. P. 14, Col 9 lines 16-19: Therefore, the authentication software 54 may be installed in an application environment in the secure area 62 such that it may obtain the authentication table **without passing through an unsecured part of the device**.	EXHIBIT 1, p.20: The Secure World is a hardware-isolated environment in which highly sensitive software executes. EXHIBIT 2, p.41: Keys stored in the TIMA KeyStore are further encrypted with the device-unique hardware key (DUHK) and can only be decrypted from within TrustZone Secure World on the same device. EXHIBIT 3, p.3: Time of design - By design, Samsung KNOX fully leverages the **hardware Trusted Execution Environment (TEE) capabilities** found in Samsung's flagship mobile devices, as well as many others. Without a TEE or equivalent, secure computing cannot be meaningfully achieved. For example, TEE uses ARM® TrustZone®. EXHIBIT 12, p.1-2: Device-side Security: Samsung Pay, TrustZone, and the TEE Worlds apart from other wallet apps Samsung's Galaxy-class devices supporting KNOX and Samsung Pay employ ARM® TrustZone® technology , a system- on-chip (SoC) security architecture that establishes two hardware-based "worlds" — a Normal World and a Secure World. The Normal World is where non-secure software and data processing takes place. **The Secure World is reserved for storage and**

		computing of sensitive (encrypted) data and the associated cryptographic keys. By erecting a strong security perimeter between the two worlds, hardware logic present in the TrustZone bus fabric prevents Normal World components from accessing Secure World resources…. **Applications that run in the Secure World are called Trusted Apps(TAs).** Shown in Figure 4, multiple TAs comprising the Samsung Pay architecture, such as those responsible for communications with the payment networks, **run inside the TEE**. There are others as well, including the trusted apps that handle user authentication and those responsible for managing data encryption and authentication keys for the Payment Framework. For user authentication, trusted drivers operating in the TEE control access to the fingerprint sensor and the touch sensor for the Trusted PIN Pad. **These drivers only allow authentication information to be passed directly to the respective payment network trusted app (for Visa, MasterCard, American Express, et al) inside the TEE.**
providing authentication software in said electronic device, the private key being accessible to said authentication software,	P.10, Col 1 lines 63-66: providing authentication software in said electronic device, the authentication data being accessible to said authentication software.	EXHIBIT 1, p.20: The Secure World is a hardware-isolated environment in which highly sensitive software executes EXHIBIT 1, p.32: Software runs in the secure world **to perform all cryptographic operations** EXHIBIT 12, p. 1-2: Shown in Figure 4, multiple TAs comprising the Samsung Pay architecture, such as those responsible for communications with the payment networks, run inside the TEE. There are others as well, including the trusted apps that handle user authentication and those responsible for managing data encryption and authentication keys for the Payment Framework. For user authentication, trusted drivers operating in the TEE control access to the fingerprint

		sensor and the touch sensor for the Trusted PIN Pad. These drivers only allow authentication information to be passed directly to the respective payment network trusted app (for Visa, MasterCard, American Express, et al) inside the TEE. EXHIBIT 13, p. 19: TEE-based security for device and data at rest • Card data and keys protected by hardware based keys • **Trusted Application for each card network**; **handles crypto logic; assists in implementing public-key cryptography and more** • Trusted PIN pad and fingerprint authentication directly against TEE • System integrity check via Secure Boot, Trusted boot and remote attestation, Verification of Trusted App and Mandatory Access Control
wherein authentication software is stored in said secure memory inaccessible to said operating system;	P. 14, Col. 9, lines. 28-34: This algorithm may be protected like the authentication table in the secure storage location 60, because if both become known to a user, the user may be able to forge a digital signature. By locking the authentication table and the authentication software 54 including the digital signing algorithm in the secure area 62 they are secured.	EXHIBIT 1, p.20: The Secure World is a hardware-isolated environment in which highly sensitive software executes EXHIBIT 1, p.32: Software runs in the secure world to perform all cryptographic operations EXHIBIT 13, p. 19: **TEE-based security for device and data at rest** • Chase Card data and keys protected by hardware based keys **• Trusted Application for each card network; handles crypto logic; assists in implementing public-key cryptography and more** • Trusted PIN pad and fingerprint authentication directly against TEE • System integrity check via Secure Boot, Trusted boot and remote attestation, Verification of Trusted App and Mandatory Access Control
activating the authentication software to generate a digital signature from the private key,	P.10, Col 1 lines 61-67: The method comprises providing authentication data in a memory of said electronic device, which authentication data are inaccessible to a user of the electronic device; providing authentication software in said electronic device, the authentication data being	EXHIBIT 10, p.2: When making a purchase, the Chase user's device sends the payment token, along with a cryptogram, to the merchant POS, which relays them to the payment network for approval by the issuer (Chase)**. The cryptogram is generated**

	accessible to said authentication software; activating the authentication software to generate a digital signature from the authentication data; providing the digital signature to the second transaction party. P. 15, Col 11 lines 65-67, Col 12 lines 1-3: Referring to Fig. 5A again, the application 50 may require a digital signature for an electronic transaction or for verification of legitimate use of digital data. Thereto, the authentication software 54 is executed to initiate decryption of the authentication data stored in memory location 63 and for generating said digital signature	**using a cryptographic key based on the algorithm furnished by the card network.** **This key, stored in the device's trusted execution environment (TEE), called TrustZone, is static or dynamic, depending on the card brand/network**. Static cryptographic keys are used over a relatively long period of time and in multiple key exchanges, whereas a dynamic key is generated for each exchange. EXHIBIT 13, p.17: • **Cryptogram is effectively a one-time use digital signature** or Verification Value (Visa) or Checksum (MC) that verifies the cryptogram from device with the one generated by TSP in the cloud.
wherein the authentication software is run in a secure processing environment inaccessible to said operating system;	P. 14, Col 9 lines 42 – 51: To prevent that the authentication data becomes accessible to a user, the authentication data should only be decrypted in a secure processing environment such as a 'Ring Zero' processing environment, which is embedded in a processing unit of commonly used personal computers. Such a secure processing environment may only be accessible to selected software applications, preferably not to user-operated software applications. Further, a decryption algorithm and/or a decryption key should not be obtainable to any user. P. 14, Col 9 lines 16-19: Therefore, the authentication software 54 may be installed in an application environment in the secure area 62 such that it may obtain the authentication table without passing through an unsecured part of the device.	EXHIBIT 12, p. 1-2: Shown in Figure 4**, multiple TAs comprising the Samsung Pay architecture,** such as those responsible for communications with the payment networks, **run inside the TEE**. There are others as well, including the trusted apps that handle user authentication and those responsible for managing data encryption and authentication keys for the Payment Framework. For user authentication, trusted drivers operating in the TEE control access to the fingerprint sensor and the touch sensor for the Trusted PIN Pad. These drivers **only allow** authentication information to be passed directly **to the respective payment network trusted app** (for Visa, MasterCard, American Express, et al) **inside the TEE.**

providing the digital signature to the second transaction party.	P.10, Col 1 lines 61-67: The method comprises providing authentication data in a memory of said electronic device, which authentication data are inaccessible to a user of the electronic device; providing authentication software in said electronic device, the authentication data being accessible to said authentication software; activating the authentication software to generate a digital signature from the authentication data; providing the digital signature to the second transaction party P. 10, Col 2 lines 31-37: In a private network access transaction, the first transaction party is uniquely identifiable by its digital signature. Said digital signature is provided to the second transaction party that may store the digital signature. If a problem arises later, the first transaction party may be traced and identified by the digital signature provided to the second transaction party.	EXHIBIT 10, p.1: Cryptographic key generation The cryptogram is another important element ensuring a transaction's integrity. It contains encrypted data derived from the token (DPAN), timestamp, and Application Transaction Counter (ATC), which are used to prevent a "replay" event — repeating a transaction using the same authorization code. **When making a purchase, the Chase user's device sends the payment token, along with a cryptogram, to the merchant POS (second transaction party), which relays them to the payment network for approval by the issuer (Chase)**. The cryptogram is generated using a cryptographic key based on the algorithm furnished by the card network. The card networks hold the master key to their card product and use it to generate a unique derived key (UDK) for each cardholder, which remains unchanged for the lifetime of the card. EXHIBIT 13, p.17: •**Cryptogram is effectively a one-time use digital signature** or Verification Value (Visa) or Checksum (MC) that verifies the cryptogram from device with the one generated by TSP in the cloud.

US 11,063,766 B2 – Date of Patent: July 13, 2021

’766 Patent Claim 1		
Feature	**Support from Patent**	**Chase Bank’s Use**
A method for performing an electronic transaction between a first transaction party and a second transaction party	P.10, Col 1 lines 65-67, Col 2 lines 1-9: At least this object is achieved in the present invention by a method for performing an electronic transaction between a first transaction party and a second transaction party using an electronic device operated by the first transaction party. The method comprises providing authentication data in a memory of said electronic device, which authentication data are inaccessible to a user of the electronic device; providing authentication software in said electronic device, the authentication data being accessible to said authentication software; activating the authentication software to generate a digital signature from the authentication data; providing the digital signature to the second transaction party.	EXHIBIT 14: The Chase cardholder is the first transaction party and the second transaction party is the merchant that the Chase cardholder is purchasing from with Chase Pay + Samsung Pay. EXHIBIT 9, p. 5-10: Chase directs Chase cardholder-users to add Chase cardholder data to Samsung Pay for mobile use in stores, when shopping online and in apps. . EXHIBIT 9, p. 13: Many Chase Visa and Mastercard credit cards, debit and Chase Liquid cards work with Samsung Pay.
using an electronic device operated by the first transaction party,	P. 14, Col. 10, lines 41-44: The authentication data are generated from a bit string that is generated at the device of the first transaction party and the bit string and is therefore not known to a second transaction party or a trusted third party (TTP).	EXHIBIT 9: the first transaction party, a Chase cardholder uses a Samsung mobile device
the electronic device having an operating system creating a run-time environment for user applications	P. 13, Col 8 lines 49-50: The operating system 48 creates a: run-time environment for any user applications.	EXHIBIT 1, p. 5: In this section, we examine the basic structure of **the Android OS,** which KNOX is built on. Recall that a hardware processor provides two modes, user and privileged. Operating systems use both of these modes for various functions. The portion running in privileged mode is **called the kernel**. OS kernels are among the most rigorously engineered pieces of software in the world, because they must perform many functions, all with the power of the processor’s

		privileged mode. For example, any time data arrives for the phone from the Internet, the OS kernel first chooses whether to even allow the data to proceed, or to drop it if it seems unwanted. If the data is allowed, **the kernel examines it and decides which application on the phone the data is intended for.** The kernel then places the data in the app's memory, and notifies the app that data has arrived. If the app then wishes to send a reply, the app's reply is sent by repeating this whole process in reverse.
and authentication software running in a separate operating environment, independent from and inaccessible to the operating system,	P. 14, Col 9 lines 58 – 67: To prevent that the authentication data becomes accessible to a user, the authentication data should only be decrypted in a secure processing environment such as a 'Ring Zero' processing environment, which is embedded in a processing unit of commonly used personal computers. Such a secure processing environment may only be accessible to selected software applications, preferably not to user-operated software applications. Further, a decryption algorithm **and/or a decryption key should not be obtainable to any user.** P. 14, Col 9 lines 32-35: Therefore, the authentication software 54 may be installed in an application environment in the secure area 62 such that it may obtain the authentication table without passing through an unsecured part of the device.	EXHIBIT 1, p.20: Most of the system as we know it, including the kernel and middleware, as well as all apps, execute in what is called the normal world. Normal world software can never access the data used by Secure World software. EXHIBIT 3, p.3: Time of design - By design, Samsung KNOX fully leverages the hardware Trusted Execution Environment (TEE) capabilities found in Samsung's flagship mobile devices, as well as many others. Without a TEE or equivalent, secure computing cannot be meaningfully achieved. For example, TEE uses ARM® TrustZone®. EXHIBIT 12, p. 1-2: Shown in Figure 4**, multiple TAs comprising the Samsung Pay architecture,** such as those responsible for communications with the payment networks, **run inside the TEE**. There are others as well, including the trusted apps that handle user authentication and those responsible for managing data encryption and authentication keys for the Payment Framework. For user

		authentication, trusted drivers operating in the TEE control access to the fingerprint sensor and the touch sensor for the Trusted PIN Pad. These drivers only allow authentication information to be passed directly to the respective payment network trusted app (for Visa, MasterCard, American Express, et al) inside the TEE.
the electronic device having a memory comprising storage locations, part of the memory being accessible to the operating system, part of the memory being a secure area,	P. 14, Col 9 lines 11-14: In or behind the console 52, there is a secure area 62 only accessible to the BIOS. The secure area 62 comprises applications and storage locations, which are not reported to the operating system 48.	EXHIBIT 3, p.7: TIMA Client Certificate Management (CCM) **<u>Enables cryptographic keys to be sequestered in a secure area of the device</u>**, so that private key information is never exposed to the Android operating system. EXHIBIT 4, p.7: TIMA KeyStore Samsung KNOX builds and improves upon the basic Android KeyStore foundation to provide enterprise class mobile application and data security with **<u>a hardware-based TIMA KeyStore. KNOX uses a TrustZone-based TIMA KeyStore for cryptographic key storage.</u>** EXHIBIT 5, p. 13: Upon device start up, the Chase user's mobile device uses the Samsung Secure Boot Key (SSBK) to check all software components. One of the components is the **<u>TrustZone Secure World, a chip partition reserved for secure code and data</u>**. Only specially privileged software modules running within the TrustZone Secure World can access these keys. EXHIBIT 5, p. 14: ARM TrustZone Secure World – the Secure World is the environment where highly sensitive software runs. **<u>The ARM TrustZone</u>**

		hardware ensures memory and **components marked secure can only be accessed in the Secure World**. Most of the system, including the kernel, middleware, and apps, run in the Normal World. The Secure World software, on the other hand, is more privileged, and can access both Secure and Normal World resources. EXHIBIT 11, p. 1: All of Chase's Samsung Pay tokens and keys are **stored within the TEE in encrypted form using a hardware-based device key that is unique to each device.** EXHIBIT 12, p.1-2: Device-side Security: Samsung Pay, TrustZone, and the TEE Worlds apart from other wallet apps Samsung's Galaxy-class devices supporting KNOX and Samsung Pay employ ARM® TrustZone® technology , a system- on-chip (SoC) security architecture that establishes two hardware-based "worlds" — a Normal World and a Secure World. The Normal World is where non-secure software and data processing takes place. **The Secure World is reserved for storage and computing of sensitive (encrypted) data and the associated cryptographic keys.** By erecting a strong security perimeter between the two worlds, hardware logic present in the TrustZone bus fabric prevents Normal World components from accessing Secure World resources.

wherein the electronic device comprises a system for accessing a memory location in the memory, wherein the system for accessing the memory location is configured to selectively report the storage locations of the secure area, wherein the storage locations of which are not reported to the operating system while at the same time being reported to the authentication software running in the separate operating environment, the method comprising:	P. 14, Col 9 lines 32-35: Therefore, the authentication software 54 may be installed in an application environment in the secure area 62 such that it may obtain the authentication table without passing through an unsecured part of the device.	EXHIBIT 5, pg. 6: The Knox Platform uses a hardware-backed trusted environment and the specific components depend on the device hardware. For example, ARM Processors provide a Trusted Execution Environment (TEE) that leverages components such as the ARM TrustZone, ARM Hypervisor Mode, and Embedded Secure Elements. Knox features that use the trusted environment include Real-time Kernel Protection (RKP), Trusted Boot, Device Health Attestation, Certificate Management, Sensitive Data Protection (SDP), and Network Platform Analytics (NPA) EXHIBIT 12, p. 1-2: Worlds apart from other wallet apps Samsung's Galaxy-class devices supporting KNOX and Samsung Pay employ ARM® TrustZone® technology , a system- on-chip (SoC) security architecture that establishes two hardware-based "worlds" — a Normal World and a Secure World. The Normal World is where non-secure software and data processing takes place. **The Secure World is reserved for storage and computing of sensitive (encrypted) data and the associated cryptographic keys.** By erecting a strong security perimeter between the two worlds, **hardware logic present in the TrustZone bus fabric prevents Normal World components from accessing Secure World resources.** Shown in Figure 4, multiple TAs comprising the Samsung Pay architecture, such as those responsible for communications

		with the payment networks, run inside the TEE. There are others as well, including the trusted apps that handle user authentication and those responsible for managing data encryption and authentication keys for the Payment Framework. For user authentication, trusted drivers operating in the TEE control access to the fingerprint sensor and the touch sensor for the Trusted PIN Pad. **These drivers only allow authentication information to be passed directly to the respective payment network trusted app** (for Visa, MasterCard, American Express, et al) **inside the TEE.**
providing a private key in the secure area of the electronic device	P. 12, Col. 6, lines 60-67: The installation in a new device is completed according to cell 20. The encoded authentication table is stored in a secure part of the BIOS, where it may only be retrieved by the BIOS and not by the operating system. This secure part of the BIOS may also be any other secure location in the device. For instance, it may be a separate part of a hard drive of a computer, which part may not be accessible to the operating system, but only to the BIOS **and/or authentication software.**	EXHIBIT 10, p.2: When making a purchase, the Chase user's device sends the payment token, along with a cryptogram, to the merchant POS, which relays them to the payment network for approval by the issuer (Chase). The cryptogram is generated using a cryptographic key based on the algorithm furnished by the card network. **This key, stored in the device's trusted execution environment (TEE), called TrustZone**, is static or dynamic, depending on the card brand/network. The card networks hold the master key to their card product and use it to generate a unique derived key (UDK) for each cardholder, **which remains unchanged for the lifetime of the card.**
which private key is inaccessible to a user of the electronic device,	P. 14, Col. 9 lines 11-14: In or behind the console 52, there is a secure area 62 only accessible to the BIOS. The secure area 62 comprises applications and storage locations, which are not reported to the operating system 48.	EXHIBIT 1, p.15: The TIMA KeyStore stores all application keys in the TrustZone Secure World storage

		EXHIBIT 1, p.20: The Secure World is a hardware-isolated environment in which highly sensitive software executes. (...) Normal world software can never access the data used by Secure World software. EXHIBIT 4, p.7: TIMA KeyStore Samsung KNOX builds and improves upon the basic Android KeyStore foundation to provide enterprise class mobile application and data security **with a hardware-based TIMA KeyStore. KNOX uses a TrustZone-based TIMA KeyStore for cryptographic key storage.** EXHIBIT 10, p.2: When making a purchase, the Chase user's device sends the payment token, along with a cryptogram, to the merchant POS, which relays them to the payment network for approval by the issuer (Chase). The cryptogram is generated using a cryptographic key based on the algorithm furnished by the card network. See Token handling by Samsung Pay for additional details. **This key, stored in the device's trusted execution environment (TEE), called TrustZone**, is static or dynamic, depending on the card brand/network.
wherein the secure area is inaccessible to the operating system of the electronic device, thereby rendering the private key inaccessible to the user;	P. 14, Col. 9 lines 11-14: In or behind the console 52, there is a secure area 62 only accessible to the BIOS. The secure area 62 comprises applications and storage locations, which are not reported to the operating system 48.	EXHIBIT 1, p.15: The TIMA KeyStore stores all application keys in the TrustZone Secure World storage EXHIBIT 1, p.20: The Secure World is a hardware-isolated environment in which highly sensitive software executes. (...) Normal world software can never access the data used by Secure World software.

		EXHIBIT 13, p.19 TEE-based security for device and data at rest • Chase Card data and **keys protected by hardware based keys** • Trusted Application for each card network; **handles crypto logic; assists in implementing public-key cryptography and more** • Trusted PIN pad and fingerprint authentication directly against TEE • System integrity check via Secure Boot, Trusted boot and remote attestation, Verification of Trusted App and Mandatory Access Control
providing authentication software in the electronic device, the private key being accessible to the authentication software,	P. 12, Col. 6, lines 60-67: The installation in a new device is completed according to cell 20. The encoded authentication table is stored in a secure part of the BIOS, where it may only be retrieved by the BIOS and not by the operating system. This secure part of the BIOS may also be any other secure location in the device. For instance, it may be a separate part of a hard drive of a computer, which part may not be accessible to the operating system, but only to the BIOS **and/or authentication software**.	EXHIBIT 1, p.20: The Secure World is a hardware-isolated environment in which highly sensitive software executes EXHIBIT 1, p.32: Software runs in the secure world to perform all cryptographic operations EXHIBIT 12, p.1-2: Shown in Figure 4, **multiple TAs comprising the Samsung Pay architecture**, such as those responsible for communications with the payment networks, **run inside the TEE.** There are others as well, including the trusted apps that handle user authentication and those responsible for managing data encryption and authentication keys for the Payment Framework. For user authentication, trusted drivers operating in the TEE control access to the fingerprint sensor and the touch sensor for the Trusted PIN Pad. These drivers only allow authentication information to be passed directly to the respective payment network trusted app (for Visa,

		MasterCard, American Express, et al) inside the TEE.
wherein the authentication software is stored in the secure area inaccessible to the operating system;	P. 14, Col 9 lines 42-50: This algorithm may be protected like the authentication table stored in the secure storage location 60, because if both become known to a user, the user may be able to forge a digital signature. By locking the authentication table and the authentication software 54 including the digital signing algorithm in the secure area 62 they are secured.	EXHIBIT 1, p.20: The Secure World is a hardware-isolated environment in which highly sensitive software executes EXHIBIT 1, p.32: Software runs in the secure world to perform all cryptographic operations EXHIBIT 13, p. 19: **TEE-based security for device and data at rest** • Chase Card data and keys protected by hardware based keys • **Trusted Application for each card network; handles crypto logic; assists in implementing public-key cryptography and more** • Trusted PIN pad and fingerprint authentication directly against TEE • System integrity check via Secure Boot, Trusted boot and remote attestation, Verification of Trusted App and Mandatory Access Control
activating the authentication software to generate a digital signature from the private key,	P.10, Col 2 lines 1-9: The method comprises providing authentication data in a memory of said electronic device, which authentication data are inaccessible to a user of the electronic device; providing authentication software in said electronic device, the authentication data being accessible to said authentication software; activating the authentication software to generate a digital signature from the authentication data; providing the digital signature to the second transaction party. P. 15, Col 12 lines 14-19,: Referring to Fig. 5A again, the application 50 may require a digital signature for an electronic transaction or for verification of legitimate use of digital data. Thereto, the authentication software 54 is executed to initiate decryption of the authentication data	EXHIBIT 10, p.2: When making a purchase, the Chase user's device sends the payment token, along with a cryptogram, to the merchant POS, which relays them to the payment network for approval by the issuer (Chase)**.** **The cryptogram is generated using a cryptographic key based on the algorithm furnished by the card network.** **This key, stored in the device's trusted execution environment (TEE), called TrustZone, is static or dynamic, depending on the card brand/network**. Static cryptographic keys are used over a relatively long period of time and in multiple key exchanges,

	stored in memory location 63 and for generating said digital signature	whereas a dynamic key is generated for each exchange. EXHIBIT 13, p.17: •**Cryptogram is effectively a one-time use digital signature** or Verification Value (Visa) or Checksum (MC) that verifies the cryptogram from device with the one generated by TSP in the cloud.
wherein the authentication software is run in a secure processing environment inaccessible to the operating system; and	P. 14, Col 9 lines 58 – 67: To prevent that the authentication data becomes accessible to a user, the authentication data should only be decrypted in a secure processing environment such as a 'Ring Zero' processing environment, which is embedded in a processing unit of commonly used personal computers. Such a secure processing environment may only be accessible to selected software applications, preferably not to user-operated software applications. Further, a decryption algorithm **and/or a decryption key should not be obtainable to any user.** P. 14, Col 9 lines 32-35: Therefore, the authentication software 54 may be installed in an application environment in the secure area 62 such that it may obtain the authentication table without passing through an unsecured part of the device.	EXHIBIT 1, p.32 Software runs in the secure world to perform all cryptographic operations. EXHIBIT 3, p.3: Time of design - By design, Samsung KNOX fully leverages the hardware Trusted Execution Environment (TEE) capabilities found in Samsung's flagship mobile devices, as well as many others. Without a TEE or equivalent, secure computing cannot be meaningfully achieved. For example, TEE uses ARM® TrustZone®. EXHIBIT 12, p. 1-2: Shown in Figure 4**, multiple TAs comprising the Samsung Pay architecture,** such as those responsible for communications with the payment networks, **run inside the TEE**. There are others as well, including the trusted apps that handle user authentication and those responsible for managing data encryption and authentication keys for the Payment Framework. For user authentication, trusted drivers operating in the TEE control access to the fingerprint sensor and the touch sensor for the Trusted PIN Pad. These drivers **only allow** authentication information to be passed directly **to the respective payment**

		network trusted app (for Visa, MasterCard, American Express, et al) **inside the TEE.**
providing by the electronic device the digital signature to the second transaction party.	P.10, Col 1 lines 65-67, Col 2 lines 1-9: The method comprises providing authentication data in a memory of said electronic device, which authentication data are inaccessible to a user of the electronic device; providing authentication software in said electronic device, the authentication data being accessible to said authentication software; activating the authentication software to generate a digital signature from the authentication data; providing the digital signature to the second transaction party P. 10, Col 2 lines 39-45: In a private network access transaction, the first transaction party is uniquely identifiable by its digital signature. Said digital signature is provided to the second transaction party that may store the digital signature. If a problem arises later, the first transaction party may be traced and identified by the digital signature provided to the second transaction party.	EXHIBIT 10, p.1: Cryptographic key generation The cryptogram is another important element ensuring a transaction's integrity. It contains encrypted data derived from the token (DPAN), timestamp, and Application Transaction Counter (ATC), which are used to prevent a "replay" event — repeating a transaction using the same authorization code. **When making a purchase,** the Chase user's device sends the payment token, **along with a cryptogram, to the merchant POS,** which relays them to the payment network for approval by the issuer (Chase). **The cryptogram is generated using a cryptographic key based on the algorithm furnished by the card network.** The card networks hold the master key to their card product and use it to generate a unique derived key (UDK) for each cardholder, which remains unchanged for the lifetime of the card. EXHIBIT 13, p.17: •**Cryptogram is effectively a one-time use digital signature** or Verification Value (Visa) or Checksum (MC) that verifies the cryptogram from device with the one generated by TSP in the cloud.

Works Cited

Exhibit 1 .... The Samsung Knox Security Solution White Paper (April 2015)
Exhibit 2 .... Samsung Knox Security Solution White Paper (May 2017)
Exhibit 3 .... Android Security Maximized by Samsung Knox
Exhibit 4 .... Beyond Basic Android Security with Samsung Knox
Exhibit 5 .... Knox Platform for Enterprise (KPE) White Paper (2019)
Exhibit 6 .... RSA Laboratories PKCS#11 Mechanisms for One-Time Password Tokens
Exhibit 7 .... An Overview of the Samsung Knox 2.0 Platform White Paper (March 2014)
Exhibit 8 .... An Overview of the Samsung Know™ Platform White Paper (August 2016)
Exhibit 9 .... Chase – Digital Payments
Exhibit 10 .... Tokenization/Samsung Developers
Exhibit 11 .... Token Handling by Samsung Pay/Samsung Developers
Exhibit 12 ...... Device-side Security: Samsung Pay, TrustZone, and the TEE/Samsung Developers
Exhibit 13 .... Samsung Pay – Presentation
Exhibit 14 .... Chase Pay and Samsung Pay Join Forces – Samsung US Newsroom

EXHIBIT 1

SAMSUNG KNOX

White Paper: Samsung KNOX™ Security Solution

April 2015
Enterprise Mobility Solutions
Samsung Electronics Co., Ltd.

SAMSUNG
BUSINESS

Contents

Section 1: BYOD and mobile security 3

Section 2: Background: What's in a smartphone? 4

Smartphone hardware 4

The Android operating system 5

Boot process 6

Mobile device security 6

Section 3: Samsung KNOX overview 7

The Samsung KNOX philosophy 7

Samsung KNOX design 8

Problem: Lack of trust in Android security 11

Solution: Base security in a hardware-rooted trusted environment 11

Build trust 12

Secure Boot and Trusted Boot 12

Maintain trust 12

Runtime protections 12

Prove trust 13

Problem: Mixing enterprise data and user apps on one device 13

Solution: Protect enterprise apps and data in a secure Workspace 13

Problem: Device theft 14

Solution: Protect enterprise Data-at-Rest by default 14

Problem: Difficulty of securely implementing custom enterprise applications 15

Solution: Provide KNOX security services to enterprise applications 15

Problem: Lack of enterprise manageability and utilities 15

Solution: Provide extensive manageability and utilities 16

Section 4: Technology in depth 18
Part 1. Building a hardware-rooted trusted environment 19
Hardware Roots of Trust 19
Establishing trust 20
Maintaining trust 22
Proving trust 28
Part 2. Making the trusted environment enterprise ready 29
SE for Android 29
Sensitive Data Protection 31
On-Devce Encryption 32
Trusted Boot Based KeyStore (TIMA KeyStore) 32
Trusted Boot Based Client Certificate Management (TIMA CCM) 32
Trusted UI 33
Data erase during factory reset 33
Section 5: Enterprise readiness 34
KNOX Workspace: Divide and conquer 34
Virtual Private Network 36
SmartCard framework 38
Single Sign-On 38
Active Directory Integration 39
Mobile Device Management 39
Simplified enrollment 41
Enterprise Billing 42
Endnotes 43
About Samsung Electronics Co., Ltd. 44

Acronyms

AES	Advanced Encryption Standard
AOSP	Android Open Source Project
BYOD	Bring Your Own Device
CAC	U.S. Common Access Card
CCM	Client Certificate Management
CESG	Communications and Electronic Security Group
CMK	Container Master Key
COPE	Corporate-Owned Personally Enabled
DAC	Discretionary Access Control
DAR	Data-at-Rest
DISA	U.S. Defense Information Systems Agency
DIT	Data-in-Transit
DoD	U.S. Department of Defense
DRK	Device Root Key
DUHK	Device-Unique Hardware Key
FIPS	Federal Information Processing Standard
IAM	Identity and Access Management
IPC	Inter Process Communication
MAC	Mandatory Access Control
MAM	Mobile Application Management
MCM	Mobile Container Management
MDM	Mobile Device Management
MMU	Memory Management Unit
NFC	Near Field Communication

Acronyms

NIST	National Institute of Standards and Technology
ODE	On-Device Encryption
OS	Operating System
PKCS	Public Key Cryptography Standards
PKM	Periodic Kernel Measurement
RKP	Real-time Kernel Protection
ROM	Read-Only Memory
RP	Rollback Prevention
SBU	Sensitive But Unclassified
SDP	Sensitive Data Protection
SEAMS	SE for Android Manager Service
SE for Android	Security Enhancements for Android
SE Linux	Security-Enhanced Linux
SRG	Security Requirements Guide
SSBK	Samsung Secure Boot Key
SSO	Single Sign-On
STIGs	Security Technical Implementation Guides
TIMA	TrustZone-based Integrity Measurement Architecture
VPN	Virtual Private Network

Section 1: BYOD and mobile security

It was only eight years ago that smartphones entered the market. Employees were already bringing their personal phones to work, but smartphones suddenly allowed access to corporate email, making it easier to respond to work-related demands after work or during business travel. Then document sharing added to the ease of doing business on mobile devices. The evolution of Bring-Your-Own-Device (BYOD) and Corporate-Owned-Personally-Enabled (COPE) started slowly, and then accelerated with the proliferation of apps for every business and personal need. While enterprise employees enjoyed the freedom and productivity of *always connected,* IT Admins were blindsided with the new and growing problem of protecting corporate intellectual property from the avalanche of unprotected personal property employees brought to work.

The security model used by IT departments was originally designed to protect an enterprise network and company-issued PCs, not the personal smartphones and tablets employees started bringing into the workplace. With both BYOD and cyber-attacks increasing, the scramble to analyze the facts and figures ensued in hopes of finding a way to manage the moving target of mobile security.

What are the numbers? What are the risks?

The Juniper Networks Mobile Threat Center, a global research facility on mobile security, released its third annual Mobile Threats Report in June 2013 from data collected from March 2012 through March 2013. They found mobile malware threats growing at a rapid rate of 614 percent to 276,259 total malicious apps, demonstrating an exponentially higher cybercriminal interest in exploiting mobile devices.[1] In another study, the 2013 Global Application Security Risk Report said 98% percent of applications presented at least one application security risk, while the average application registered 22.4 risks.[2]

Mobile malware
is growing
at a rapid rate of
614%
~
276,259
malicious
apps

Apps aren't the only security threat in the mobile landscape, but they are the biggest threat. A Nielsen February 2014 report, The Digital Consumer, reported that smartphone owners spend 86% of their time using apps versus the mobile web.[3] And report after report pointed to the real culprit for lack of mobile security as the Android open source code that hackers could easily use to create and distribute malicious apps.

While these facts and figures were being reported, Samsung was already at work designing solutions for the problems. In 2012, a group of Samsung engineers set out to build a trusted environment rooted in the hardware of the Android operating system (OS) that could be used cross-platform for any other OS. And, the blueprint included maintaining the trusted platform, and providing management and tools for an enterprise ready mobile security solution. In 2013, Samsung KNOX became available to enterprises large and small, giving them complete control over how they implemented their security configuration, and assurance that Samsung KNOX is a trusted mobile security solution always evolving to meet customer needs.

Section 2: Background: What's in a smartphone?

An understanding of the inner workings of an Android smartphone is helpful in appreciating the many security measures in Samsung KNOX. This section provides an overview of how Android-based phones work. Readers already familiar with Android, and mobile architectures, including ARM® TrustZone®, may wish to skip ahead to the next section.

There is much more to a smartphone than the mere apps and widgets a user typically experiences. Behind the scenes is a highly sophisticated system of advanced processor architectures, operating system kernels, libraries, and middleware and security related services. The following three sections aim to demystify each of these concepts, paving the way for a firm understanding of the security capabilities of Samsung KNOX.

Smartphone hardware

Just like a desktop computer, at the heart of every mobile device, are one or more processor cores. These are the central computational unit of the device, where all code for the phone's apps and operating system runs. The processor is also physically connected to the phone's many hardware devices. These include the antennas used for LTE, and Wi-Fi, and the internal storage drives, as well as any removable SD cards and docking ports.

One of the most important features of a processor is the use of modes of execution. A mode defines how much privilege a piece of software running on the processor has. For example, when user-installed apps run on the processor, it runs in user mode. In user mode, the apps are not allowed to directly access hardware devices or resources controlled by other apps. On the other hand, when critical operating system software is running, the processor is in privileged mode. In privileged mode, the system's software is allowed to directly access hardware devices, as well as all data held by the user's applications. Clearly, any code running in privileged mode must be protected from control by adversaries (attackers, malicious users, etc.).

KNOX leverages a processor architecture known as ARM TrustZone. While TrustZone maintains the two modes described above, it also provides a new security-specific construct called worlds. In TrustZone, there are two worlds, the normal world, and the Secure World. Virtually all smartphone software as we know it today still runs in the normal world. The Secure World is reserved for highly-sensitive computations such as those involving cryptographic keys (see the Mobile Security section below). As described throughout this document, KNOX makes extensive use of TrustZone's Secure World, both for protecting enterprise confidential data, and for monitoring the OS kernel running in the normal world. Given these highlights of the TrustZone processor architecture, the next section explains two more security critical components, the Android OS, and its kernel.

The Android operating system

In this section, we examine the basic structure of the Android OS, which KNOX is built on. Recall that a hardware processor provides two modes, user and privileged. Operating systems use both of these modes for various functions. The portion running in privileged mode is called the kernel. OS kernels are among the most rigorously engineered pieces of software in the world, because they must perform many functions, all with the power of the processor's privileged mode. For example, any time data arrives for the phone from the Internet, the OS kernel first chooses whether to even allow the data to proceed, or to drop it if it seems unwanted. If the data is allowed, the kernel examines it and decides which application on the phone the data is intended for. The kernel then places the data in the app's memory, and notifies the app that data has arrived. If the app then wishes to send a reply, the app's reply is sent by repeating this whole process in reverse.

Given this example, consider what could happen if an attacker gained control of the OS kernel. Due to the kernel's high permissions, the attacker could tamper with and leak arbitrary sensitive data from any application, and send it to anywhere on the Internet. This is why KNOX implements the extensive protections for OS kernels covered in later sections.

In most traditional operating systems, when applications wish to communicate with each other, they ask the kernel to set up the lines of communication for them. To facilitate more rich and easy to use forms of app communication, the Android OS instead provides another layer of software, fittingly called the middleware.

The Android middleware runs in user mode, but sits between the kernel and apps. The middleware provides a rich set of communication methods that allow apps to share their data with each other and perform operations on each other's behalf. For example, many image library apps offer the option to take photos, even though they don't know how to use the phone's camera. Instead, they simply request that an app that does understand the camera take the picture on their behalf. A second major function of KNOX is to ensure that such forms of communication do not occur between enterprise apps and user apps. This prevents sensitive data from being leaked to an untrusted third-party, and prevents corrupted data from entering enterprise apps.

Boot process

An important concept that ties together the hardware, kernel, and apps is the boot process. When a device is first turned on, the user's applications are not immediately available. Instead, a chain of software components start, with each component starting the next one in the chain. Typically, when the user presses the ON button, the device first runs a program called a boot loader. Many mobile device architectures use multiple levels of bootloaders to perform different functions. The boot loader then finds where the kernel is stored, and begins running the kernel in the processor's privileged mode. The kernel starts the Android middleware and some basic apps, running them in user mode. At this point, the user is presented with a login screen, and can begin using the phone.

Mobile device security

This final section explains some important concepts in the security of mobile devices. An significant detail that differentiates mobile security from other domains is that device owners have complete control over how to use their own devices, as opposed to, say, a corporate-owned laptop, which is controlled by IT administrators. For example, in a BYOD scenario, sensitive emails are likely be downloaded to the device, but the user may simultaneously compromise the kernel's security to allow for their own device customizations. This process is typically known as device rooting.

Even though users' motivations for rooting are often benign, such as installing custom themes, the subsequent security breach makes it easier for malicious parties to gain control of the device. The same techniques have been known to be used by malware authors as such access can steal the user's data or use their device to attack others. Many of the security measures put in place by KNOX are designed to either prevent device rooting, or to mitigate the resulting damage.

Another fundamental feature for mobile device security is the use of cryptography. KNOX uses cryptography for three key functions:

- **Encryption** - the scrambling of data using a protected key to keep it confidential
- **Hashing** - the creation of a unique series of numbers to represent a particular piece of software or data; a single difference in a piece of data yields a different hash.
- **Signing** - the encryption of a hash of a piece of data using a private key to prove that the data originated from a particular party

KNOX frequently uses signing to produce signatures of hashes of firmware components. This proves that the firmware component originated from the owner of the private key used for signing, and in this case, it proves the component originated from Samsung. KNOX maintains signing keys are only accessible in the TrustZone Secure World.

Section 3: Samsung KNOX overview

Enterprise data is increasingly finding its way onto smartphones as a result of BYOD and COPE policies.

This new way of working has increased productivity for employees by placing work and personal data, such as emails, on the same device. However, this has also greatly complicated the task of IT security. Mobile devices provide numerous avenues through which sensitive data can fall into the wrong hands, such as sharing of data with untrusted third-party applications, device theft, intentional rooting by power users, and misconfigured or vulnerable enterprise applications. These problems are manifest across organizations with hundreds or thousands of devices, all requiring security management and configuration.

In this whitepaper, we present Samsung KNOX. KNOX aims to be the most comprehensively secure and manageable mobile device solution for enterprises large and small. Based on the Android OS, Samsung KNOX is designed around the philosophy that the foundations of device security should be rooted in fixed hardware mechanisms. KNOX bases this foundation in the principles of trusted computing, a set of methods for making devices that can prove to enterprises they are running the correct security software, and can raise alerts in the event that tampering is detected. On top of this trusted foundation, KNOX builds a Workspace environment to protect enterprise apps and their data, a robust set of data at rest protections, and a large suite of enterprise security tools, including a highly configurable Virtual Private Network (VPN) and Mobile Device Management (MDM) interfaces.

We begin our overview with the design philosophy that is behind everything we have built in KNOX.

The Samsung KNOX philosophy

KNOX is built using a two-step design philosophy:

Step 1. Build a trusted environment rooted in hardware security mechanisms.
In a trusted environment, sensitive or enterprise-critical functionality is only enabled once the device is in an allowed state. Here, state refers to the security-relevant software and configurations on the device. For example, parts of state considered by KNOX include the bootloaders, kernel, and TrustZone OS, as well as security policy configurations. Furthermore, the trusted environment allows for remote attestation, where any change can be securely summarized via a set of proofs. Third parties can then inspect these proofs to decide if the device state meets their security requirements. A trusted environment is considered hardware rooted if its security is based on the high difficulty of physically tampering with hardware circuits.

Why is it important to root trust in hardware? The designers of KNOX are aware that throughout the history of operating system security, certain critical functionalities have always been trusted to privileged system software (mainly the OS kernel). However, in the last two decades, attackers have become more and more successful at exploiting kernel flaws whether on their own or others' devices. Other mechanisms such as heavyweight virtual machines or special Basic Input/Output System (BIOS) checks have been implemented and circumvented. The KNOX design recognizes that to date, the single best defense against full-system compromise is to tie any system self-checks to a secret maintained by secure hardware, which is out of the reach of any software-based adversary, and virtually all physically present adversaries as well.

Step 2. Make the trusted platform ready for enterprise use.
The trusted platform must be usable by enterprises. This involves giving enterprises complete control and configurability over their data. The KNOX Workspace protects enterprise data using encryption, and the enterprise manages the Workspace using Mobile Device Management (MDM) capability. In addition, KNOX supplies a collection of useful secure applications and utilities, such as VPN, that allow enterprise-ready deployment. The security of the KNOX Workspace and utilities is firmly grounded in the hardware root of trust described in the first step, and in isolating the Workspace from the personal space.

Samsung KNOX design

Through this two-step design philosophy, KNOX addresses the most pressing security problems facing enterprises' BYOD and COPE strategies today. To this end, we identify the following key challenges in making an Android-based system enterprise ready:

1. The problem of device rooting
2. Mixing enterprise data with user apps on the same device
3. Device theft
4. Difficulty of securely implementing custom enterprise applications
5. Lack of enterprise manageability and utilities

The following sections detail each of these problems, followed by the solution stemming from the KNOX design philosophy.

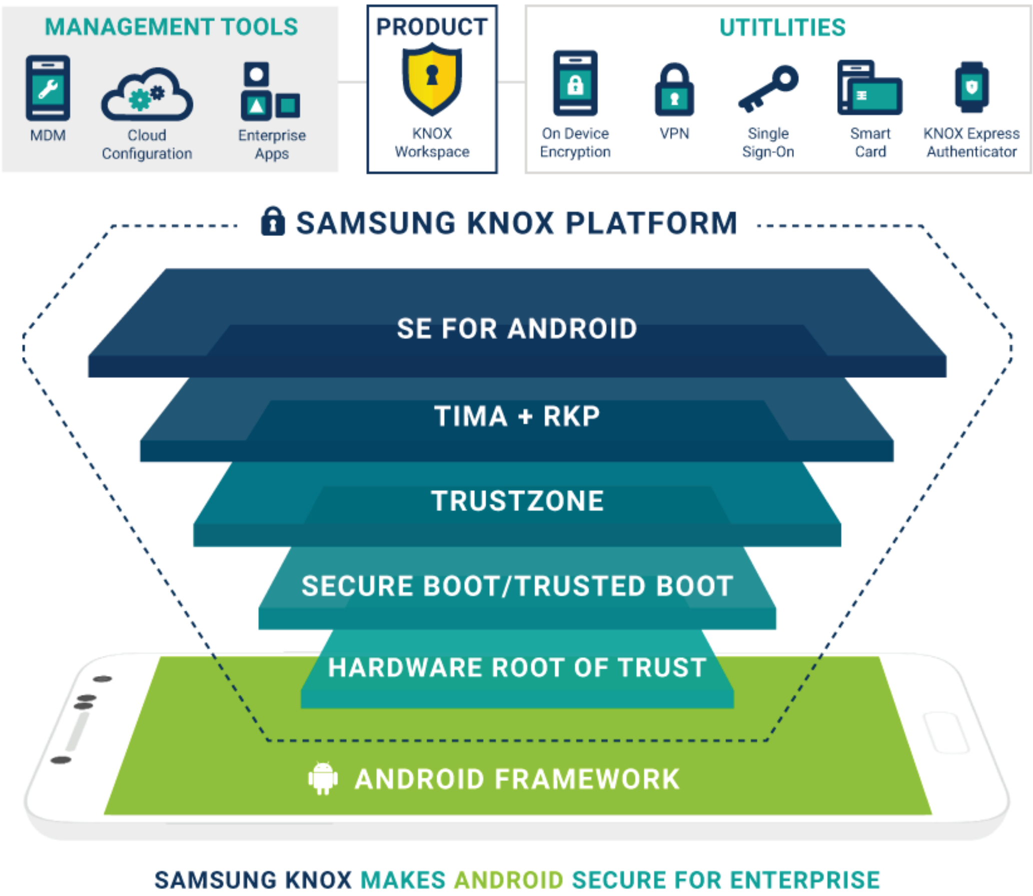


Figure 1 – Samsung KNOX Enterprise Security Solution

Table 1 summarizes the structure of the following section by pairing each of the above listed hurdles to enterprise adoption of Android with the KNOX-specific solution.

KNOX Strategy	Problem(s) Solved	Solution Technologies
Build a Hardware-Rooted Trusted Environment	Lack of trust in Android security	**Hardware Root of Trust** Samsung Secure Boot Key, Rollback Prevention Fuses, KNOX Warranty Bit, Device Root Key (DRK) **Build Trust** Trusted Boot using TrustZone-Based Integrity Measurement Architecture (TIMA), Rollback Prevention **Maintain Trust** Real-Time Kernel Protection (RKP), Periodic Kernel Measurement (PKM), DM-Verity **Prove Trust** TIMA Attestation
Make Trusted Environment Enterprise-Ready	Mixing enterprise data and user apps on one device	KNOX Workspace, Security Enhancements for Android
	Device theft	KNOX Workspace Encryption, Sensitive Data Protection (SDP), On-Device Encryption (ODE)
	Difficulty of securely implementing custom enterprise applications	TIMA KeyStore, Client Certificate Manager (CCM), SE for Android Management Service (SEAMS)
	Lack of enterprise manageability and utilities	Mobile Device Management (MDM), Virtual Private Network (VPN), Active Directory Integration, Single Sign-On (SSO)

Table 1 - KNOX Solution Technologies

Problem: Lack of trust in Android security

The Android OS was originally designed for end users and not for enterprise use. The original Android approach to security was to simply isolate apps from each other. However, this alone is insufficient to provide confidence for enterprise use. For example, how can enterprises be confident that security measures are even enabled, when it is common practice for users to root their devices? (Device rooting is the practice of intentionally exploiting privileged software to circumvent vendor-included restrictions.)

Solution: Base security in a hardware-rooted trusted environment

KNOX provides strong guarantees for the protection of enterprise data by building a hardware-rooted *trusted environment*. A trusted environment ensures that enterprise-critical operations, such as decryption of enterprise data, can only occur when the device is proven to be in an allowed state. For many pieces of device software, such as the kernel and TrustZone apps, the *allowed* state is represented by the cryptographic signature of each piece of software. A trusted environment is hardware-rooted if both the cryptographic keys and code used to compute these signatures are tied back to unmodifiable values stored in hardware. KNOX facilitates a hardware-rooted trusted environment in three steps.

- Build trust by enforcing that only approved versions of system-critical software are loaded
- Maintain trust by ensuring that system-critical software is not modified once loaded
- Prove that only approved system-critical software is loaded and run on a particular device when requested to do so by the enterprise.

Figure 2 below shows how KNOX builds, maintains, and attests its trusted environment.

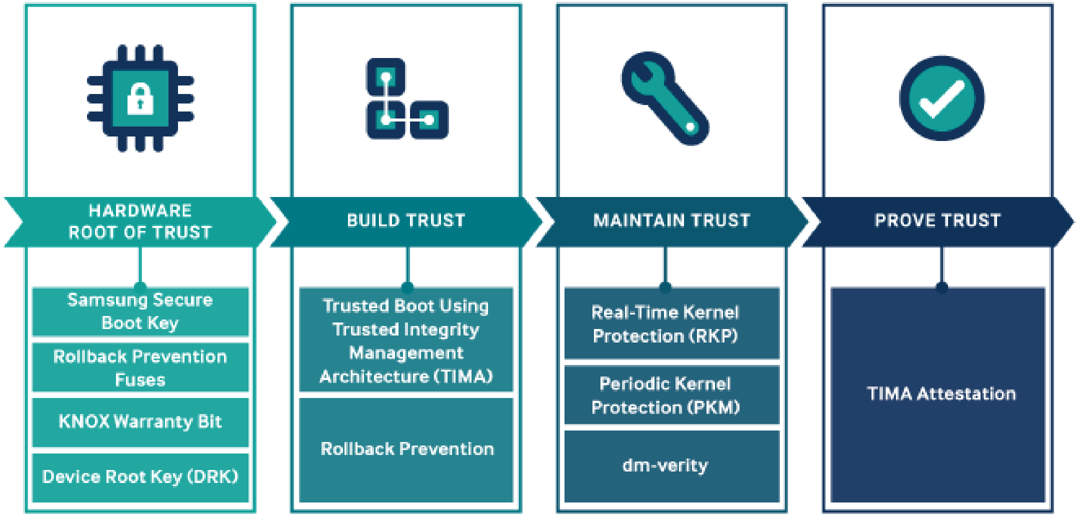


Figure 2 - KNOX Builds, Maintains and Attests Its Trusted Environment

Build trust

Secure Boot and Trusted Boot

When a device is first turned on, the user's applications are not immediately available. Instead, a chain of software components is started, with each component starting the next one in the chain. Typically, once the hardware is powered on, it first runs a program called a bootloader, which in turn runs the operating system's kernel, a highly privileged component that starts applications and can access storage and network devices directly.

Many device vendors support a process known as Secure Boot, and Samsung KNOX devices are no exception. In a Secure Boot process, each component in the boot chain (bootloader, kernel, etc.) checks the integrity of the next component through signature verification. If the signature verification fails, the boot process is stopped.

Secure Boot is limited because it cannot distinguish between different approved versions, for example, a bootloader with a known vulnerability and a later patched version, since both versions have valid signatures. To address this limitation, Samsung KNOX adopts Trusted Boot in addition to Secure Boot. In the Trusted Boot process, each software component in the chain measures and securely stores the cryptographic hash of the next component in TrustZone Secure World memory before loading it. Storing these measurements allows a third-party to identify the exact versions of software loaded on the device through the process of attestation. For example, this can be used to verify that only the latest patched versions of software are run, complementing the Rollback Prevention feature that ensures patched software is not downgraded to a vulnerable version.

If signature verification fails, KNOX either records the tampering by blowing a one-time fuse, called the KNOX warranty fuse, or by preventing further booting, depending on the configuration. Devices that have the fuse set cannot run certain KNOX features such as the KNOX Workspace thereafter.

Both Secure Boot and Trusted Boot have their trust rooted in hardware. The first piece of software loaded is the primary boot loader, which is kept in hardware-protected Read-Only Memory (ROM). In addition, the cryptographic key used to verify signatures is the Samsung Secure Boot Key, also stored in hardware fuses.

Maintain trust

Runtime protections

At the end of a successful secure boot, only approved versions of system software (such as the TrustZone OS) have been loaded. However, they can then be modified. For example, users may either intentionally or unintentionally run code that exploits a flaw to maliciously modify the kernel, thus bringing it under adversarial control. KNOX detects kernel compromises quickly using a pair of techniques: Real-time Kernel Protection (RKP) to actively prevent kernel code modification, and Periodic Kernel Measurements (PKM) that periodically check kernel code integrity. Both RKP and PKM checks are performed from the TrustZone Secure World.

The kernel is not the only attractive target for malware and malicious users. There are large numbers of other code objects and configurations that can be used by malware to become persistent, meaning that it can restart itself each time the device restarts. KNOX prevents such modifications by integrating Google's DM-Verity, a kernel module that verifies the integrity of applications and data stored on the critical system partition. In the event a malicious process or user modifies something on the system partition, DM-Verity detects the modification the next time the data is read, and blocks any attempted access to modified data.

Prove trust

Consider an MDM server that wishes to interact with a mobile device. The MDM should not simply assume that the device is uncompromised. Instead, a KNOX-enabled device provides the MDM with an attestation, a cryptographically verifiable collection of device state measurements. This includes hashes of the bootloaders, kernel, TrustZone OS, and logs from runtime protection mechanisms, among others. (For the complete contents of the attestation message, see TIMA Attestation in Section 4: Technology in depth.) The MDM can then decide if all entries on the list are approved. The attestation is signed using a key derived from the Device Root Key (DRK), which is hardware protected. Thus, if we trust the hardware to be untampered and the ARM TrustZone Secure World software to work properly, a trusted environment is established, and this can be proved to a third party.

The problems in subsequent sections are solved using technologies built on top of this trusted environment. We stress that without the trusted environment covered in this section, all remaining security measures are ineffective, as there is no guarantee that they were even loaded and run as expected.

Problem: Mixing enterprise data and user apps on one device

With the emergence of BYOD and COPE, one of the major challenges facing enterprises is the mixing and interaction of sensitive enterprise apps and data with potentially malicious user-installed apps on the same device. Android provides apps with many ways to interact with one another. Apps may share databases containing photos or contact information, and perform actions on each other's behalf, such as a browser opening a link in an SMS text message. This design has greatly benefited the mobile ecosystem, resulting in an explosion of useful applications. However, this ease of sharing can be problematic where data from sensitive enterprise emails and documents can easily be leaked to untrusted apps that claim to provide a necessary functionality. Enterprises must have solid guarantees that their data is safe even with hundreds or thousands of employees downloading untrustworthy third-party apps.

Solution: Protect enterprise apps and data in a secure Workspace

To solve this problem, we needed to provide strong isolation between the enterprise and user aspects of the device. Such strong isolation guarantees are provided by Mandatory Access Controls (MAC). In a MAC system, access to resources is restricted using a policy that can only be modified by the device vendor, in this case, Samsung.

Therefore, we decided to adapt the Security Enhancements for Linux (SELinux) MAC system for KNOX. SELinux provides a rich policy language for describing fine-grained access to resources by programs. We extend SELinux into Security Enhancements for Android (SE for Android). SE for Android provides additional mediation locations in KNOX's Android middleware, along with additional policy language. KNOX's pioneering of SE for Android has now led to its adoption into the Android Open Source Project (AOSP).

The KNOX Workspace is built on top of SE for Android to define a protected environment for enterprise data and apps. The Workspace provides a full environment, including the home screen, launcher, applications, and widgets. This environment runs alongside the user's environment, but it is protected from interference from user-installed applications. All data created by container applications is kept on a protected partition as described in the following section. In the event that tampering is detected during Trusted Boot, the container and its data are no longer accessible.

Problem: Device theft

Smart phone theft is one of the most serious threats to the confidentiality of enterprise data. A 2014 study, “Smart phone thefts rose to 3.1 million last year, Consumer Reports finds,” estimated that only 7% of smartphone users enable encryption for Data at Rest. Furthermore, only 36% enable a screen lock, and 34% take no security precautions at all.[4] Also, a recent FCC study cites FBI data estimating that nearly 10% of all thefts and robberies in the US in 2013 were related to theft of a mobile device.[5] Given the high prevalence of mobile device theft, and the reluctance of users to secure their data in the event of theft, a secure mobile OS must protect data without user initiative.

Solution: Protect enterprise Data-at-Rest by default

Samsung KNOX does not depend on users to secure their own BYOD devices in case of loss or theft. KNOX defines two classes of data – *protected* and *sensitive*. All data written by apps in the secure Workspace is considered protected. Protected data is encrypted on disk when the device is powered off. In addition, the decryption key for protected data is tied to the device hardware. This makes protected data recoverable only on the same device. Furthermore, access controls are used to prevent applications outside the KNOX Workspace from attempting to access protected data.

Even stronger protection is applied to *sensitive* data. Sensitive data remains encrypted as long as the Workspace is locked, even if the device is powered on. When the user unlocks their KNOX Workspace using their password, Sensitive Data Protection (SDP) allows sensitive data to be decrypted. When the user re-locks the Workspace, SDP keys are cleared. The SDP data decryption key is tied to both device hardware and to the user input. Therefore, the data is recoverable only on the same device and with user input.

SDP can be used in one of two ways. First, all emails received are considered sensitive, and are immediately protected by SDP encryption. Emails received when the Workspace is locked, are immediately encrypted, and can only be decrypted the next time the Workspace is unlocked. The second way to use SDP is through the KNOX Chamber. The Chamber is a designated directory on the file system. Any data placed into the Chamber is automatically marked as sensitive and protected by SDP.

Problem: Difficulty of securely implementing custom enterprise applications

Properly implementing cryptographic, authentication, and secure storage services has traditionally been challenging. Vulnerabilities are regularly found in both cryptographic libraries and applications that leak keys. Once a key is leaked, all data previously encrypted with that key becomes vulnerable. Keeping secret keys secret is a problem for a number of reasons. First, many applications make multiple copies of keys in their internal logic, which they do not properly track and delete, thus increasing the risk of key leakage. Aggravating the problem, implementation flaws, such as the now infamous Heartbleed bug, can allow secret keys to be leaked directly to the network. In spite of the risks associated with implementing cryptographic services in applications, many enterprise apps require them.

Solution: Provide KNOX security services to enterprise applications

KNOX allows enterprise developers to build custom applications on top of its hardware-rooted trusted environment. KNOX exposes APIs for key management, and FIPS 140-2 compliant cryptographic algorithms. The Trusted Boot-based TIMA KeyStore provides applications one type of secure key storage. Recall that Trusted Boot only allows sensitive operations to occur if approved versions of all security-critical system software are loaded. The TIMA KeyStore stores all application keys in the TrustZone Secure World storage. From there, the keys can only be accessed if Trusted Boot is successful. Thus, an application's keys are safe, even in the event that a user or malicious app tampers with critical system software.

Similar to TIMA KeyStore, Client Certificate Management (CCM) is a complementary service for generating, storing and using asymmetric key pairs and certificates in the TrustZone Secure World storage. The CCM API provides applications with PKCS#11 compliant token management, as well as public key algorithms for signatures and encryption.

Problem: Lack of enterprise manageability and utilities

KNOX introduced the functionality and manageability required for enterprise use. First, to use the secure environment effectively, enterprises need utilities such as VPN and Microsoft Exchange integration. Second, enterprises need complete control to configure the Workspace to meet their security needs. We realized that each enterprise balances security and functionality differently. For example, enterprises have widely differing password policies. Further, enterprises have lists of approved applications, identity providers, and IT partners they trust to deliver their IT infrastructure.

Solution: Provide extensive manageability and utilities

Enterprises large and small have very diverse sets of needs when it comes to device management. Samsung KNOX gives enterprises complete control to configure the Workspace to their needs using an extensive set of more than 1500 Mobile Device Management (MDM) APIs.

Second, Samsung KNOX provides utilities that allow ready deployment in enterprises. Samsung KNOX offers per-application VPN controls, a smartcard framework, and Single Sign-On (SSO) integration with Microsoft Active Directory. These features enable Samsung KNOX to easily integrate into any enterprise.

MDMs can obtain proof that their devices are running a trusted environment using TIMA attestations. The TIMA attestation data contains the cryptographic hashes of the boot components and other critical security information such as if SE for Android is enabled. This data is signed using a key derived from the DRK, which proves that the attestation data originated from the TrustZone Secure World on a particular Samsung device.

For large enterprises requiring extensive customization above what is offered by MDMs, Samsung KNOX has a dedicated team to determine unique needs and prepare custom KNOX flavors.

Finally, to build enterprise confidence, Samsung KNOX has obtained a number of certifications:

FIPS 140-2 Certification

Issued by the National Institute of Standards and Technology (NIST), the Federal Information Processing Standard (FIPS) is a US security standard that helps ensure companies that collect, store, transfer, share, and disseminate sensitive but unclassified (SBU) information and controlled unclassified information (CUI) can make informed purchasing decisions when choosing devices to use in their workplace.

Samsung KNOX meets the requirements for FIPS 140-2 Level 1 certification for both data-at-rest (DAR) and data-in-transit (DIT).

DISA Approved STIG

The Defense Information Systems Agency (DISA) is an agency within the US DoD that publishes Security Technical Implementation Guides (STIGs) which document security policies, requirements, and implementation details for compliance with DoD policy.

On April 17, 2014 DISA approved the STIG for Samsung KNOX 1.0.

DISA Approved Product List

On May 14, 2014 DISA added five KNOX-enabled devices to the US DoD Approved Products List (APL).

***NOTE**: The five Samsung devices added to DISA's APL are the only Android 4.4 OS devices on the list as of May 14, 2014. They are also the only devices certified under Common Criteria (CC) by Mobile Device Fundamental Protection Profile (MDFPP). These five new devices represent a twenty percent increase of mobile products available for purchase in the DoD.*

Common Criteria Certification

The Common Criteria for Information Technology Security Evaluation, commonly referred to as Common Criteria, is an internationally-recognized standard for defining security objectives of information technology products and for evaluating vendor compliance with these objectives. A number of Governments use Common Criteria as the basis for their own certification schemes.

Select Galaxy devices with KNOX embedded received Common Criteria (CC) certification on February 27, 2014. The current CC certification targets the new Mobile Device Fundamentals Protection Profile (MDFPP) of the National Information Assurance Partnership (NIAP), published in October 2013, which addresses the security requirements of mobile devices for use in enterprise.

CESG Approved

The Communications and Electronic Security Group (CESG) approved KNOX-enabled Android devices for United Kingdom government use on May 14, 2014.

FICORA

FICORA: Samsung devices with KNOX fulfill national security requirements as defined by the Finnish National Security Auditing Criteria (KATAKRI II).

ASD

Australian Signals Directorate: ASD endorsing the Protection Profile for Mobile Device Fundamentals as well as recognizing evaluations against this Protection Profile.

NIAP-validated

Samsung KNOX is approved by the United States government as the first NIAP-validated consumer mobile devices to handle the full range of classified information.

For the most up-to-date list of certifications, please visit https://www.samsungknox.com/en/security-certifications .

Section 4: Technology in depth

The KNOX design philosophy consists of two steps:

1. Build a hardware-rooted trusted environment. A trusted environment ensures that sensitive and enterprise-critical operations only occur once the device is proven to be in an allowed state. Part 1 examines how KNOX builds a hardware-rooted trusted environment and how it can prove to a third-party that it runs a trusted environment.

2. Make the trusted environment enterprise ready. KNOX provides enterprise security services such as VPN, secure storage, cryptographic APIs and secure isolation of enterprise apps from untrustworthy user apps, just to name a few. Part 2 delves into each of these, relating each back to the trusted environment provided in Part 1.

Subsequent sections detail the two steps. Figure 3 is an overview of the KNOX design.

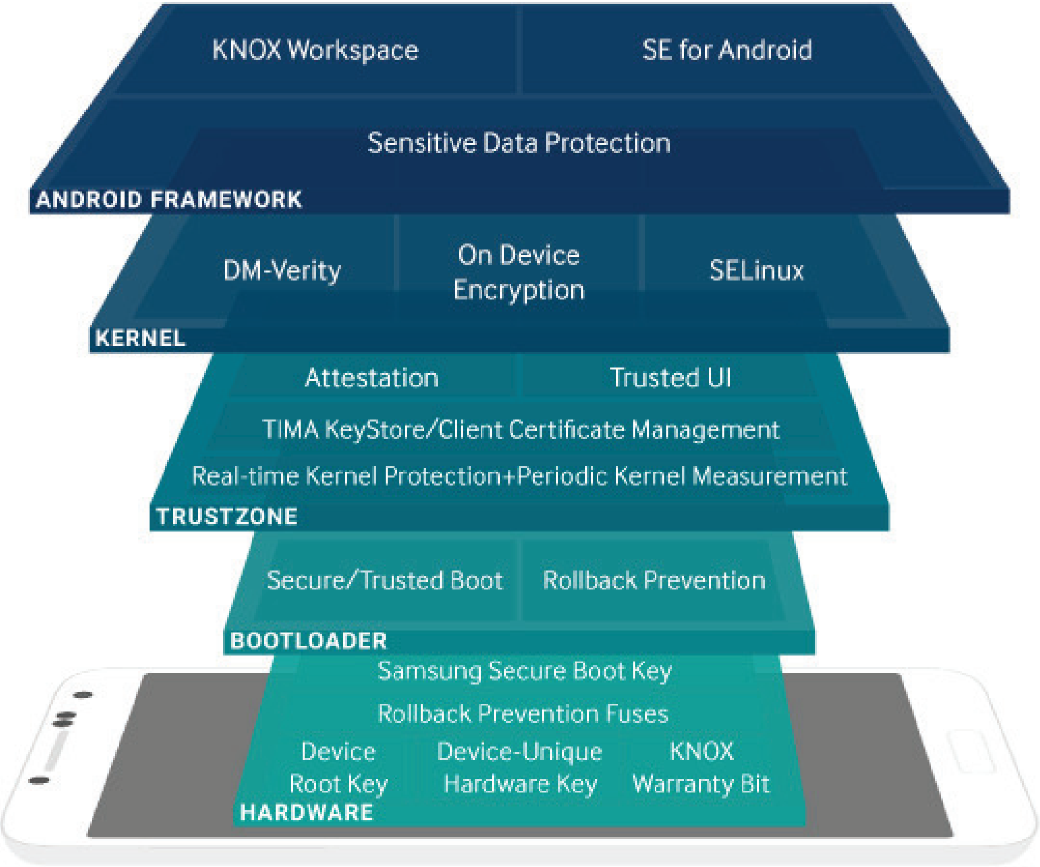


Figure 3 - KNOX Architecture Overview

Part 1. Building a hardware-rooted trusted environment

This first part examines how KNOX builds its trusted environment in four subsections. First, we examine the hardware roots of trust, which trust in all other components relies upon. Second, we present how KNOX establishes trust during boot time. Third, we show how KNOX maintains the already established trust while the system is in use. Finally, we examine how KNOX proves its trustworthiness to remote parties such as the enterprise management system.

Hardware Roots of Trust

In this section, we describe the hardware components that are the foundation of Samsung KNOX's trusted environment.

Device-Unique Hardware Key (DUHK). The DUHK is a device-unique symmetric key that is set in hardware at manufacture time in the Samsung factory. The DUHK provides a way to bind data to a particular device as follows. The DUHK is only accessible to a hardware cryptography module and is not directly exposed to any software. However, software can request for data to be encrypted and decrypted by the DUHK. Data encrypted by the DUHK becomes bound to the device, because it cannot be decrypted on any other device. The DUHK is typically used to encrypt other cryptographic keys.

Samsung Secure Boot Key (SSBK). The SSBK is an asymmetric key pair used to sign Samsung-approved executables of boot components. The private part the SSBK is used by Samsung to sign the secondary and application bootloaders. The public part of the SSBK is stored in hardware one-time programmable fuses at manufacture time in the Samsung factory. The Secure Boot process uses this public key to verify whether each boot component it loads is approved (See the section on Secure Boot).

Rollback Prevention Fuses (RP Fuses). The RP fuses are hardware fuses that encode the minimum acceptable version of Samsung-approved bootloaders. Old software may contain known vulnerabilities that can be exploited. The rollback prevention feature prevents approved, but out-of-date versions of bootloaders from being loaded (See the section on Rollback Prevention). The version number in the RP fuses is set when system software is first installed and later during updates. RP fuses are one-time programmable. Thus, the minimum acceptable version can only be incremented but not decremented.

KNOX Warranty Fuse. The KNOX warranty bit is a one-time programmable fuse that signifies whether the device has ever been booted into an unapproved state. If the Trusted Boot process detects that non-approved components are used, or if certain critical security features such as SELinux are disabled, it sets the fuse. Thereafter, the device can never run Samsung KNOX, access to the DUHK and DRK in the TrustZone Secure World is revoked, and the enterprise's data on the device cannot be recovered.

ARM TrustZone Secure World. The Secure World is a hardware-isolated environment in which highly sensitive software executes. The ARM TrustZone hardware enforces that memory and devices that are marked secure can only be accessed in the Secure World. Most of the system as we know it, including the kernel and middleware, as well as all apps, execute in what is called the normal world. Normal world software can never access the data used by Secure World software. The Secure World software, on the other hand, is more privileged, and can access both secure and normal world resources. KNOX makes extensive use of the Secure World both for cryptographic operations, and for monitoring normal world security.

Bootloader ROM. The primary bootloader (PBL) is the first piece of code to be run during the boot process. The PBL is trusted to start the measurement and verification of the boot chain (see sections on Secure Boot and TIMA Trusted Boot). To prevent tampering, the PBL is kept in secure hardware Read Only Memory (ROM). The device hardware loads and runs the PBL from ROM at boot, and the PBL starts the Secure and Trusted Boot processes.

Device Root Key (DRK). The DRK is a device-unique asymmetric key pair that is signed by Samsung's root key through an X.509 certificate. This certificate proves that the DRK was produced by Samsung. The DRK is generated at manufacture time in the Samsung factory and is stored on the device encrypted by the DUHK, thus binding it to the device. The DRK is only accessible from within the TrustZone Secure World.

Because the DRK is device-unique, it can be used to tie data to a device through cryptographic signatures. The DRK is not used directly to sign data; instead, signing keys are derived from the DRK. As an example, the TIMA attestation data, which proves the device is in a trusted state, is signed using the Attestation Key, which is itself signed by the DRK. The DRK signature proves that the attestation data originated from the TrustZone Secure World on a Samsung device. Note that while the DRK is not stored directly in hardware, it is an important part of the root of trust as it derives other signing keys, and is protected by both the DUHK and TrustZone Secure World.

Establishing trust

Android begins the startup process with the primary bootloader, which is loaded from ROM. This code performs basic system initialization and then loads another bootloader, called a secondary bootloader, from the file system into RAM and executes it. Multiple secondary bootloaders may be present, each for a specific task. The boot process is sequential in nature with each secondary bootloader completing its task and executing the next secondary bootloader in the sequence, finally loading the application bootloader known as aboot, which loads the Android operating system. This sequence of components is called the boot chain.

Secure Boot

In the Secure Boot process, each component in the boot chain verifies the integrity of the subsequent component against a signature before executing it. If verification fails, the boot process is stopped. Signatures of boot components are generated at build time using the Samsung Secure Boot Key (SSBK). The public part of the SSBK is stored in hardware fuses during manufacture. The first component in the chain, the primary bootloader, is stored in immutable ROM and is trusted to verify the secondary bootloader. Thus, the Secure Boot chain can only be compromised by hardware tampering. Later boot components, such as the kernel, are signed by another Secure Boot Key that is programmed into the previous boot component.

TrustZone-based Integrity Measurement Architecture

Secure Boot prevents the device from starting if unapproved boot components are detected. However, if the device does start, Secure Boot cannot inform a third party about what versions of approved boot components have been loaded and run. For example, it cannot distinguish between a boot component with a known vulnerability vs. a later patched version, since both versions have valid signatures. In addition, some carriers decide to allow custom OS kernels to run on their devices. On these devices, Secure Boot cannot prevent unapproved kernels from running. This clearly poses a threat to enterprise applications and data. To improve upon this limitation of Secure Boot, KNOX contains the TrustZone-based Integrity Measurement Architecture (TIMA). TIMA introduces two features: Trusted Boot and Attestation.

TIMA Trusted Boot

In Trusted Boot, each boot component in the boot chain measures the subsequent component and stores the measurement before executing it. An illustration of Trusted Boot is given in Figure 4. The measurement is a SHA256 cryptographic hash of the boot component. These hashes are securely stored in TrustZone-protected memory. The set of hashes consists of one or more secondary bootloaders, the TrustZone Secure World operating system, the application bootloader and the normal world kernel. Also, depending on the processor make and model, hashes of additional firmware images such as the modem are included. These hashes can then be used to prove the integrity of a device to a remote server through TIMA Attestation.

When unauthorized boot components are loaded, Trusted Boot will react in one of two ways. Low level components that are tightly tied to the device hardware, such as the bootloaders, should never be replaced; therefore, any attempt to replace these components produces a screen telling the user to take the device to a service center.

However, if the kernel has been modified, Trusted Boot instead sets the KNOX warranty violation fuse. This one-time programmable memory fuse indicates that the device has been tampered with and cannot use certain KNOX features thereafter. Additionally, even if the boot code is restored to its original factory state, this evidence of tampering remains and is reflected in the attestation results. Note that some device models will opt to never set the warranty violation fuse, instead always requesting that the user take the device in for service.

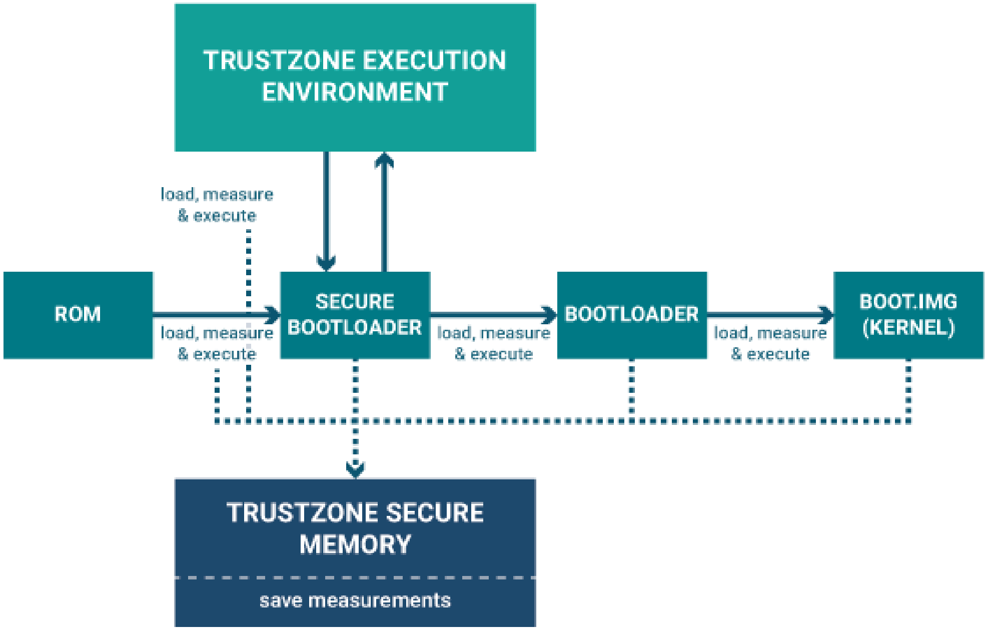


Figure 4 - The Trusted Boot Process

As each boot loader measures and executes the next, the measurements are stored in TrustZone secure memory for later inspection, i.e., through attestations.

Rollback Prevention (RP)

Rollback Prevention blocks the device from loading or flashing an approved but old version of boot components. Old versions of software may contain known vulnerabilities that attackers can exploit. Rollback prevention checks the version of the bootloader and kernel during both boot and updates, and blocks these processes from continuing if versions are unacceptably old. The lowest acceptable version of the bootloader is stored in secure hardware fuses when the device is flashed, and the lowest acceptable version of the kernel is stored in the bootloader itself. Whenever a vendor-applied update occurs, the lowest acceptable version can be incremented in the fuses. Because this value is kept in fuses, it cannot be decremented even through physical tampering.

Maintaining trust

Periodic Kernel Measurement (PKM)

TIMA PKM performs continuous periodic monitoring of the kernel to detect if legitimate kernel code and data have been modified by malicious software. In addition, TIMA also monitors key SE for Android data structures in OS kernel memory to detect malicious attacks that corrupt them and potentially disable SE for Android.

Real-time Kernel Protection

The security of the kernel is essential to the security of the whole system. An attack that compromises the kernel has the ability to arbitrarily access system sensitive data, hide malicious activities, escalate the privilege of malicious user processes, change the system behavior or simply take control of the system. As mentioned previously, Trusted Boot measurements can be used to determine what kernel was loaded and run when the device was started. However, this protection does not guarantee the integrity of the kernel after the system runs and starts to interact with potential attackers. Clever attackers can sometimes exploit an already booted and running kernel. In such cases, it is important to continuously monitor the kernel during the system *runtime* in order to detect and prevent modifications to the kernel code or critical data structures.

Intuitively, the kernel protection mechanism cannot itself exist completely in the kernel, or it could be circumvented by an attacker. Therefore, Samsung KNOX introduces Real-time Kernel Protection (RKP), a unique solution that provides the required protection using a security monitor located within an isolated execution environment. Depending on the device model, this isolated execution environment is either the Secure World of ARM TrustZone or a thin hypervisor that is protected by the hardware virtualization extensions. RKP's Trusted Computing Base (TCB) is part of this isolated environment and thus is secure from attacks that may potentially compromise the kernel.

Running in an isolated execution environment may cripple the ability of the security protection mechanisms to closely monitor events that happen inside the target kernel. To solve this problem, RKP uses special techniques to take full control over the normal world memory management and intercept critical events and inspect their impact on security before allowing them to get executed. Hence, RKP complements TIMA-PKM's periodic kernel integrity checking, which has limited effectiveness against attacks that can take place and properly hide their traces between periodic checks.

TrustZone-RKP achieves three important security features:

- First, it completely prevents running unauthorized privileged code (i.e., code that has the kernel privilege) on the system, which is accomplished by preventing modification of the kernel code, injection of unauthorized code into the kernel, or execution of the user space code in the privileged mode.

- Second, it prevents kernel data from being directly accessed by user processes. This includes preventing double mapping of physical memory that contains critical kernel data into user space virtual memory. This is an important step to prevent kernel exploits that attempt to map kernel data regions into malicious processes where they could be modified by an attacker.

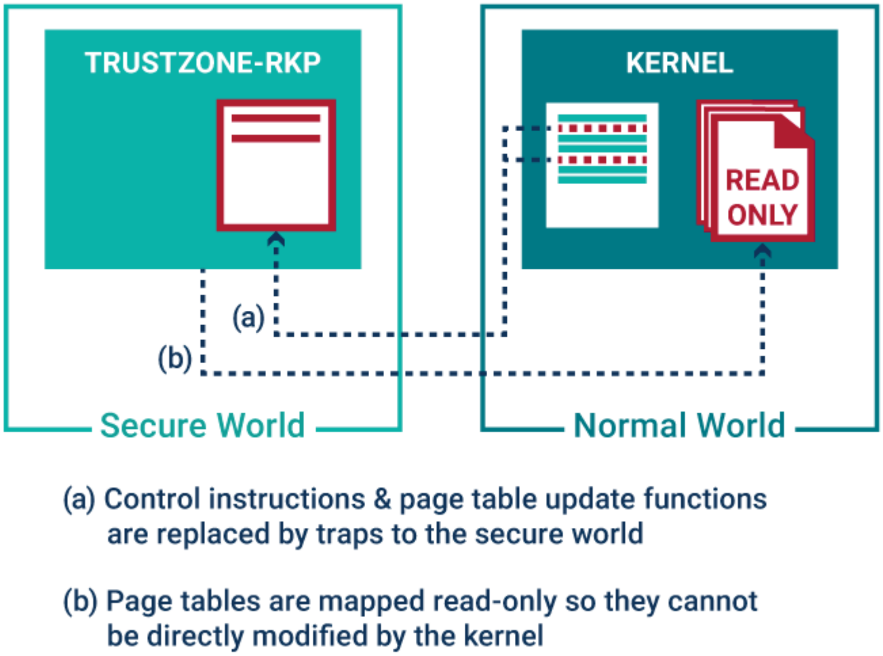
- Third, RKP monitors some critical kernel data structures to verify that they are not exploited by attacks. In particular, RKP protects the data that defines the credentials assigned to running user processes to prevent attackers from escalating this credential by modifying this data.

NOTE: *The first feature is present on all models, while the second and third features are available on select models. Additional protection features are under development.*

Architecture overview

Figure 5 shows the architecture of RKP, which is hosted in an isolated execution environment that is protected even if Android's Linux kernel is compromised. The kernel is forced it to request RKP to perform two operations on its behalf: (1) emulating control instructions that change the system state and (2) updating the normal world memory translation tables. This monitoring is enforced by depriving the kernel of its ability to control these critical functions.

System control instructions allow the normal world to control security critical system state, such as defining the location of memory translation tables and exception handlers. These instructions can be only executed by privileged code, such as the kernel code. RKP instruments the kernel so that certain system control instructions are removed from its executable memory, which is the only memory that can execute privileged instructions in the normal world. Therefore, the only way to execute these instructions is through emulating them from the Secure World. We call this operation Control Instruction Emulation. On models that use the virtualization extensions, intercepting system control instructions can also be done using the hardware virtualization extensions.



(a) Control instructions & page table update functions are replaced by traps to the secure world

(b) Page tables are mapped read-only so they cannot be directly modified by the kernel

Figure 5 - RKP Architecture (On select models, RKP runs in the virtualization protected environment rather than TrustZone Secure World).

Memory translation tables (also called page tables), define the virtual to physical address mapping and the access permissions of virtual memory. If the kernel attempts to change the current memory layout of the system through modifying translation tables, then RKP inspects these changes to confirm that they do not impact the system security. RKP ensures that translation tables cannot be modified by the normal world through making them read-only to the normal world kernel. Hence, the only way for the kernel to update the translation tables is to request these updates from RKP. As a result, RKP guarantees that this interception is non-bypassable.

Kernel code protection

Kernel code protection is the main security feature that RKP provides. The main guarantee is that an attacker that can get past the Linux kernel defenses would not be allowed to modify the kernel executable code, which greatly reduces the impact of kernel attacks on the whole system. To achieve this objective, RKP examines memory translation table modifications to enforce a set of rules that guarantee that the kernel is not writable by any code in the normal world. These rules also guarantee that the RKP monitoring cannot be bypassed even if an attacker finds a way to break the normal world kernel protections. Thus, the kernel is not able to modify its own code, even if it is compromised.

The rules are:

1. Kernel code pages are never mapped writable
2. Kernel data pages are never mapped executable
3. All memory translation tables are mapped read-only to the normal world
4. No double mappings of kernel code or any memory translation tables is allowed (Double mapping happens when the same physical memory is mapped to multiple virtual memory addresses, which might allow two different parts of the system to access the same memory with different permissions)
5. All mapped memory regions should have the Privileged eXecute Never (PXN) permission, with the exception of the OS kernel

The first two rules guarantee that the initial image of the kernel, as measured by Trusted Boot, cannot be directly modified by any potential attacker, unless it changes the memory mapping of the system by modifying the memory translation tables. This feature is still true even if the attacker takes control of the kernel itself. The rest of the rules guarantee that the memory translation tables themselves cannot be modified by the kernel, unless it sends a request to RKP. When such a request is sent, RKP verifies that the memory translation table modification does not violate the above rules. Combining these two sets of rules together, the kernel is not modified without RKP's knowledge.

These protections still require a basic assumption that the system memory management state has not been modified. Modifying the memory management system state (e.g., through changing the effective memory translation tables' base address or disabling the virtual memory protection completely) may allow an attacker to bypass the RKP monitoring. Thus, RKP uses the **Control Instruction Emulation** feature explained above to inspect these events to guarantee that they do not tamper with its monitoring.

In models that use the virtualization extensions, system control instructions are forced to trap into RKP through hardware controls. In models that use the TrustZone-based solution, this feature is complicated by the fact that TrustZone is not capable of trapping changes to the normal world state. Hence, RKP instruments the kernel to remove all instances of these system control instructions. Since these instructions can only run from privileged code, and RKP grants that privilege exclusively to the measured and protected kernel code, then it is absolutely impossible for the normal world to run these instructions without trapping to RKP. In turn, RKP validates the values to be written to the system control instructions to guarantee that they do not invalidate its kernel code protection assumptions.

Preventing double mapping of kernel data

Kernel data structures are critical to the security of the system. Maliciously modifying kernel data can lead to wide range of damage from privilege escalation to user process hiding. Since RKP completely protects the kernel code base and prevents return-to-user attacks using the PXN protection of user pages, there are only two possible methods to exploit kernel data. The first is through double mapping the memory hosting kernel data into the address space of the malicious process. The second is to alter the kernel control flow so that it maliciously modifies its own data (such as using pointer manipulation or pointer overflow).

The first class of attacks, double mapping of kernel data to malicious user processes, is a real threat to the kernel. For instance, a real-world Android exploit used an integer overflow to trick the kernel into mapping a huge range of the physical memory into the address space of the attacking process.

To prevent malicious double mapping of the kernel data, RKP ensures that physical memory pages hosting this data are not mapped to user space processes. They can only be mapped as privileged pages that cannot be accessed by the user space. RKP enforces this rule using its control of the normal world memory translation. RKP rejects any page table modification that maps kernel data to user space. To handle a related problem, RKP makes sure that no executable kernel pages are ever double-mapped to be writeable, and vice versa.

RKP relies on the target kernel to inform it about the location of its critical data. RKP embeds hooks inside the kernel code so it is informed whenever a new memory area is going to be allocated to the kernel. It then prevents this memory from being double mapped to writable memory anywhere else in the system.

This protection is effective against attacks that use double mapping to exploit kernel data. Although RKP relies on the kernel to inform it about the allocated data memory areas, this dependency does not weaken the protection. The kernel is assumed to be secure when it sends the information to RKP because this happens before the data pages are allocated. Afterwards, RKP prevents the data from being modified, except by the kernel itself.

Protecting the kernel data that defines user process credentials

After preventing kernel code modifications and double mapping of kernel data, the last class of attacks that threatens the kernel security is to alter the kernel control flow so that it maliciously modifies its own data. These attacks may include pointer manipulation, pointer overflow or return-oriented attacks.

Although RKP cannot fully protect against this class of attacks, it implements a novel technique to mitigate their effect through protecting selective kernel data structures that are critical to the system security. The data structure of choice is the process credentials data structure, which define the privilege level of the user processes running inside the device. User processes represent different running applications, such as user apps. In Linux, there is an instance of the credentials structure that is associated with each running process. This is frequently the target of rooting attacks, as by modifying this, a normal process can elevate its privilege.

RKP implements a three-step solution to protect the credential structure from malicious modifications. First, RKP makes all instances of the credential data structure read-only through controlling the memory translation tables. Second, it instruments the kernel so that all writes to the credential structures would be routed through RKP. This is guaranteed by the fact that the kernel now would not be able to write to this data from within the normal world. Before writing to the credential data, RKP examines the values to be written to make sure that they do not maliciously escalate the privileges of their corresponding user process. Determining if a user process is legitimately entitled to an escalated privilege, such as the administrative privilege, is done through combining multiple techniques. For instance, RKP prevents processes that start with regular user privilege from escalating their privilege after they start. In another example, processes that are started by applications that interface with potential attackers, such as zygote and adb shell, are not allowed to have an escalated privilege. Finally, RKP adds a check to the kernel security hooks to verify that a credential structure actually belongs to the read-only memory protected by RKP before it is effectively used to determine the privilege of the user process. Hence, it is guaranteed that a potential attacker cannot forge a malicious instance of the credential structures that is not monitored and verified by RKP.

For detailed information on RKP and the TrustZone-based implementation of RKP, see the following link on the ACM Digital Library website: http://dl.acm.org/citation.cfm?id=2660267.2660350&coll=DL&dl=GUIDE&CFID=629439201&CFTOKEN=91386218

DM-Verity

Attackers may not only be interested in attempting to modify bootloader or kernel images. There are many other software binaries and configuration files in storage which provide malware the property of persistence. Persistent malware is able to restart itself each time the system is rebooted. It does this by modifying programs or configurations on the system partition, which contains the system binaries, Android framework, and configuration files, that are started during boot. Once inserted into the boot path, the malware can survive system reboots. Additional problems can arise from tampering with system data and configurations, such as the granting of excessive privileges to vulnerable applications.

To prevent unauthorized modifications to the system partition, KNOX integrates a customized implementation of DM-Verity, a Linux/Android kernel module that performs integrity checks on all data blocks contained in a block device (such as a partition). In stock Android, DM-Verity uses a hash tree to perform integrity checks of individual data blocks. The root of the hash tree is signed by an RSA key. Whenever a data block is read into memory, DM-Verity computes the hash of the block, and then uses it, along with the other hashes on the path to the root to compute the root hash. If this computed root hash matches the signed version, the block is considered good. Otherwise, unauthorized modification of the block is detected, and the access to the data block is restricted.

KNOX's implementation of DM-Verity differs from stock Android in supporting file-based firmware over-the-air (FOTA) software updates. This approach is easier to support with the existing infrastructure than the stock block-based approach.

Proving trust

TIMA Attestation

TIMA Attestation allows a device to attest facts about its state to a remote server, such as an MDM server. The attestation message contains state measurements that can be evaluated by a server, which can then decide whether to trust the device or not. This message contains:

- Measurements collected by Trusted Boot to prove that only approved system software was loaded during boot
- Security violation logs from PKM and RKP since the last reboot
- Status of the KNOX warranty violation fuse
- Device-identifying information such as the IMEI and Wi-Fi MAC address
- A locally-computed verdict whether the device believes it is in a trustworthy state

The full attestation message is computed in the ARM TrustZone Secure World, and thus is accurate even if the entire normal world OS is compromised. Part of this attestation message is the verdict. Only when a) the measurements collected by Trusted Boot match known good values, and b) the warranty violation fuse is intact, the verdict is set to Yes to indicate attestation passed. The known good measurement values are kept in a file called tima_measurement_info, which is kept in TrustZone secure storage. This file is generated at build time. To simplify the logic of remote servers, they can directly use the verdict instead of verifying all measurements themselves.

To ensure unforgeability, the attestation message is signed using the TIMA Attestation Key, which is traceable to Samsung's root key. Each Samsung device supporting TIMA attestation has a unique RSA key pair, the device root key (DRK). The DRK is generated during manufacture, is traceable to Samsung's root key using X.509 certificates, and is stored in TrustZone. The remote server can verify the integrity of this message using Samsung's root key. To ensure that the attacker of a compromised device cannot replay old valid attestation messages, the signature includes a server-generated cryptographic nonce, which is a random number used only once.

To illustrate the use of this capability, consider the MDM server example again. Depending on the attestation verdict and the data, any further action is determined by the enterprise's MDM security policy. The security policy might choose to detach from the device, erase the contents of the secure Workspace, ask for the location of the device, or any of many other possible security recovery procedures.

Part 2. Making the trusted environment enterprise ready

Section 1 described how KNOX built a trusted environment where the integrity of its components is tied back to hardware. The subsequent sections describe technologies built on top of this trusted environment to enable KNOX for enterprise use.

SE for Android

Samsung KNOX adopts Security Enhancement for Android (SE for Android), which adds Mandatory Access Control (MAC) to Android. Many people are aware of Discretionary Access Control (DAC) mechanisms, such as Android permissions or Linux owner/group/world permissions. DAC mechanisms have limited security benefits since the user or process generating data has discretion to change the access rules for that data. A user can make bad decisions with data, which may then be leaked publicly. MAC is designed to let security experts enforce rules that can't be maliciously or ignorantly overridden by device users or software developers. Since these rules are mandatory, and cannot be altered by users or developers, they provide a way to prevent malicious code or untrusted users from accessing sensitive data or programs. MAC can be used to lock down data that a user wants to keep secret, and prevents developers from maliciously or accidentally compromising the system components that protect our devices.

SE for Android provides two layers of MAC protection:

1. **Kernel-level protection**: Android inherits the SELinux MAC abilities directly from Linux. SELinux provides MAC for kernel system calls. SELinux policy can enforce which objects these system calls can target. For example, you can specify in SELinux policy that only system-signed processes can read files in the directory/data/security. This level of control is possible because access check hooks are inserted inside the kernel. These hooks query the security policy before each system call to determine if it represents an allowed action. SELinux policies can prevent processes from reading or tampering with data, bypassing security mechanisms, or otherwise interfering with other processes. They also limit the damage from malicious or flawed programs.

2. **Android middleware protection**: There are many parts of the Android system that don't leverage system calls to get things done. An example would be the Android Intents used to start apps. This layer of software above the kernel, but below user space applications, is called the Android middleware. Additional hooks have been added to key decision points to extend MAC control to the middleware. This is known as Middleware MAC (MMAC). MMAC can enforce security policies among inter-component communication for Android Apps.

The main security objectives of SE for Android include strong data and application isolation, confining the permissions of system processes running as root, and protecting applications.

Scope of Access Control

Samsung's custom version of SE for Android provides the following unique features:

- MAC on APIs (control who can call your APIs)
- KNOX Workspace isolation of personal & business data
- On-the-fly Workspace creation for customizing your security
- Quick-response policy updates (no carrier-approved firmware update required to plug many vulnerabilities)
- Strong application isolation beyond Android's standard access control
- Extensible MAC for new KNOX features

Samsung also built an innovative global policy validation system that can detect when prohibited actions are attempted. This gives us unique visibility into how our devices are used and can alert us to new threats. This system can be used to refine our policy and very accurately grant only the permissions needed.

SE for Android Policy

SE for Android includes a set of security policy configuration files designed to meet common, general-purpose security goals. Out of the box, Samsung KNOX provides a policy that is designed to strengthen the core Android platform and meet enterprise needs. Samsung KNOX also offers the SE for Android Manager Service (SEAMS), which provides management APIs that allow enterprise IT admins to manage SE for Android. Management tasks include gathering access logs, resetting file security labels, mapping applications to different security domains, getting type context information, and getting status information about packages and Workspaces.

For enterprise apps that run in a Workspace, Samsung KNOX provides policies to enforce the isolation of application Workspaces. For example, Samsung KNOX has created new security domains and can also now enforce Multiple Category Security (MCS) isolation. Categories are used to isolate applications and data into security groupings, independent of what security domain they are assigned. Categories can then be used to ensure that personal applications and business applications with the same security domain have their access rights limited to their own areas. New Workspaces can be created on the fly without having to edit the security policy by simply applying a new security category to a group of apps.

Sensitive Data Protection

KNOX can enforce two classes of protection for data generated from within the KNOX Workspace: protected data and sensitive data. All data generated from within the KNOX Workspace is considered to be protected. Protected data residing in storage is always encrypted, and is thus protected against offline attacks, e.g., forensic analysis on a flash memory image extracted from a stolen device. Furthermore, access controls are used to prevent applications outside the KNOX workspace from attempting to access protected data. The decryption key for protected data is stored encrypted by the device-unique hardware key (DUHK). Therefore, the key is only recoverable on the same device.

Sensitive data, on the other hand, provides an even stronger security guarantee. Like protected data, sensitive data is always encrypted when on disk. Additionally, the data remains encrypted as long as the Workspace is locked. The key used to encrypt sensitive data on disk is recoverable only if the user enters the Workspace password, PIN, or pattern. Thus, if a device is stolen, the key cannot be extracted from anywhere on the device. As with protected data, the stored key material is encrypted by the DUHK, thus binding it to the device.

Enforcement of this guarantee for sensitive data is performed by KNOX Sensitive Data Protection (SDP). SDP creates a Container Master Key (CMK) that can only be decrypted with user input. If desired, the MDM (see the section Mobile Device Management) can also be used to unlock the CMK, thus preventing total data loss in the event of a forgotten Workspace password. Once the Workspace is locked, SDP clears all keys in memory after a configurable timeout (five seconds by default). In addition, SDP also flushes sensitive file data from the OS kernel's disk caches if the file is not in use by a Workspace application.

Any sensitive data received when the Workspace is locked will still be protected by SDP. This works by using a public key algorithm in which the private part of the key is maintained in an encrypted partition, and the public part is used to encrypt the new sensitive data. Once the Workspace is unlocked, the data is decrypted with the private key, and re-encrypted using the usual symmetric key, which is guarded by the CMK. Currently, email subjects, bodies and attachments are marked sensitive. Additionally, the SDP Chamber provides a directory, in which all files are automatically marked as sensitive, and protected by SDP.

On-Device Encryption

The KNOX platform further strengthens the full-device encryption capability offered by the Android platform. In addition to Android's kernel-level full device encryption, KNOX ties the encryption key to a secret maintained in trusted hardware. This feature is available only if the enterprise IT administrator activates encryption via the MDM. TrustZone-based on-device encryption (ODE) also enables enterprises to ensure that all device data is protected in the unlikely event that the operating system is compromised. While this feature is low overhead, providing system-wide encryption means less flexibility in supporting separate security levels for user and enterprise data, hence the inclusion of the finer-grained protected and sensitive data classes.

Trusted Boot Based KeyStore (TIMA KeyStore)

The TIMA KeyStore provides applications with services for generating and maintaining cryptographic keys. The TIMA KeyStore is only enabled if the Trusted Boot measurements match the known good values in the file tima_measurement_info, and if the KNOX warranty fuse is not set. Thus, cryptographic operations with keys in the KeyStore can only occur if the system was booted into an approved state. Keys stored in the TIMA KeyStore are further encrypted with the device-unique hardware key (DUHK), and can only be decrypted from within TrustZone Secure World on the same device. All cryptographic operations on the keys are performed within TrustZone Secure World.

The TIMA KeyStore has the same API as the familiar Android KeyStore APIs. Therefore, the only modification necessary is to specify that the TIMA KeyStore be used to provide the service.

Trusted Boot Based Client Certificate Management (TIMA CCM)

The TIMA CCM enables storage and retrieval of digital certificates, as well as encryption, decryption, signing, and verification in a manner similar to the functions of a SmartCard. The certificates and associated keys are encrypted with a device-unique hardware key that can only be decrypted from code running within TrustZone.

TrustZone-based CCM also provides the ability to generate a Certificate Signing Request (CSR) and the associated public/private key pairs in order to obtain a digital certificate. A default certificate is provided for applications that do not require their own certificate.

Programming interfaces for certificate storage and management are provided in the KNOX Premium SDK. Application developers are provided with industry standard PKCS #11 APIs for certificate management, and therefore interact with the CCM as if it were a virtual SmartCard. Like the TIMA KeyStore, TIMA CCM operations are permitted only if the device was booted into an approved state.

Trusted UI

Another service required by many enterprise applications is some form of authentication. For enterprises wishing to use PIN-based authentication, KNOX provides the Trusted UI for secure credential entry. The *Trusted UI* uses ARM TrustZone to create a dedicated path through hardware from the screen and keyboard to the Secure World. Any credentials entered while this path exists will be completely inaccessible to normal world programs and untrusted peripherals. Once the credentials are held in the Secure World, they are passed back to the enterprise application that initiated the authentication.

Data erase during factory reset

The factory reset procedure for Samsung devices restores device software to its original manufacturer settings. This is done before changing device ownership of a device or when disposing of a device. A critical aspect of factory reset is securely erasing existing user data so that no data is recoverable after the reset.

Erasing data on flash storage, as used in Samsung devices, requires extra care. Samsung devices store data on a type of flash storage called embedded multimedia cards (eMMC). eMMC firmware uses translation tables that map device-visible logical memory to flash physical memory to improve performance and card life. This means devices cannot reference physical flash memory directly, and thus cannot ensure that data is erased without support from the eMMC itself.

Samsung devices use several features supported by Samsung-manufactured eMMC chips to ensure data erase during factory reset. First, when the user initiates factory reset, the reset code instructs the eMMC firmware to discard the entire range of physical memory corresponding to the logical memory storing user data. Accessing discarded user data thereafter returns zeros when accessed by the device OS. Second, the Samsung eMMC controller firmware code responsible for discarding the physical memory is itself protected against malicious updates.

Note that Samsung KNOX Workspace data is always stored encrypted in flash memory, offering yet another layer of defense-in-depth. The cryptographic keys used to encrypt KNOX Workspace data are themselves stored encrypted by the device-unique hardware key, accessible only by a separate secure processor.

Section 5: Enterprise readiness

KNOX Workspace: Divide and conquer

Samsung KNOX Workspace is a defense-grade dual persona container product designed to separate, isolate, encrypt, and protect enterprise data from attackers. This work/play environment ensures work data and personal data are separated and that only the work container is managed by the enterprise. Personal information such as photos and messages are not managed or controlled by the IT department. Once activated, the KNOX Workspace product is tightly integrated into the KNOX platform.

Workspace provides this separate environment within the mobile device, complete with its own home screen, launcher, applications, and widgets.

Applications and data inside Workspace are isolated from applications outside Workspace, that is, applications outside Workspace cannot use Android inter-process communication or data-sharing methods with applications inside Workspace. For example, photos taken with the camera inside Workspace are not viewable in the Gallery outside Workspace. The same restriction applies to copying and pasting. When allowed by IT policy, some application data such as contacts and calendar data can be shared across the Workspace boundary. The end user can choose whether to share contacts and calendar notes between Workspace and personal space; however, IT policy ultimately controls this option.

The enterprise can manage Workspace like any other IT asset using an MDM solution; this container management process is called Mobile Container Management (MCM). Samsung KNOX supports many of the leading MDM solutions on the market. MCM is affected by setting policies in the same fashion as traditional MDM policies. Samsung KNOX Workspace includes a rich set of policies for authentication, data security, VPN, e-mail, application blacklisting, whitelisting, and so on.

The KNOX 2.X platform features the elimination of application wrapping, which was used by KNOX 1.0 and many other competing solutions. This is achieved by leveraging technology introduced by Google in Android 4.2 to support multiple users on devices. It reduces the barrier to entry for independent software developers wishing to develop and deploy applications for KNOX Workspace.

KNOX Workspace can also be configured for container-only mode. In this mode, the entire device experience is restricted to the Workspace. This mode is suitable for industries such as health care, finance, and others who provide devices for employees that seek to restrict access to business applications.

Workspace also has a two-factor authentication process. The user can configure the Workspace to accept a fingerprint as the primary authentication factor for the container with a PIN, password or pattern as a second factor.

The KNOX platform also supports two containers, thus meeting the needs of professionals that use their own devices for corporate use (BYOD) and have multiple employers, such as doctors or consultants.

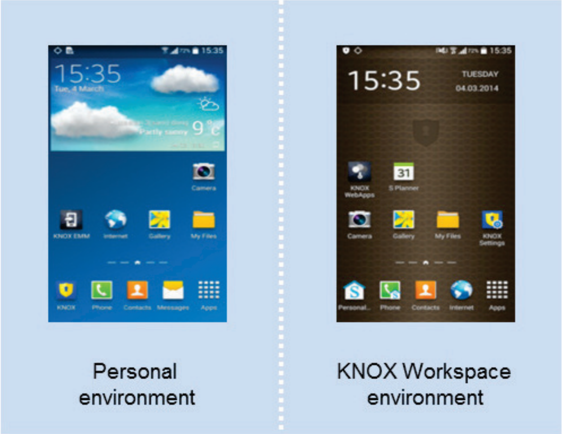


Figure 6 - User's personal environment running next to the Workspace environment

New features include the ability to enable Bluetooth® and Near Field Communication (NFC) inside Workspace. NFC enables a device to act as a SmartCard-based credential for use cases such as physical access and access to IT accounts. Bluetooth can be used to communicate with connected devices, and supports Bluetooth profiles that enable use cases beyond music and calls inside the KNOX Workspace. Examples include printing, file sharing, and external card readers. External SD cards can also be enabled with security restrictions.

KNOX caller ID for incoming calls when in Personal mode can also be configured by IT Admins to display caller ID information derived from personal contacts and KNOX Workspace contacts.

KNOX Active Protection (KAP)

End users can activate or deactivate KNOX Active Protection (KAP) via the Smart Manager app on devices not managed by an MDM. KAP uses both Real-time Kernel Protection (RKP) and DM Verity, a feature that provides integrity checking for system code and data. On MDM-managed devices, KAP is always enabled.

KNOX Quick Access

On Samsung Galaxy S6 devices, based on proximity of a registered and connected Gear device, KNOX Quick Access extends the unlock period of the KNOX Workspace, thereby reducing the frequency with which the end user must enter password credentials.

Virtual Private Network

KNOX provides a rich set of VPN features to address a wide-range of enterprise mobile device deployment scenarios. At the core of KNOX VPN framework is the Generic VPN Service that enables VPN vendors to provide a wide range of features and configurability.

VPN features of KNOX include:

- Administrator-configured System VPN
- Administrator-configured Per-App VPN
- Administrator-configured Workspace VPN
- Multiple concurrent VPN connections
- IPsec and SSL VPN support
- Administrator-configured FIPS and non-FIPS VPN mode
- Common Access Card (CAC)-based authentication
- Always on VPN connections with auto-reconnect
- VPN tunnel chaining

VPN connections in KNOX are managed by MDMs through GenericVPNPolicy class. This class provides APIs for an MDM Admin application to configure VPN settings, certificates, and applications.

KNOX VPN partners implement the KNOX Vendor SDK APIs to enable rich management capability for MDMs. A number of leading VPN vendors have released VPN clients for KNOX.

In addition to traditional VPN clients, other network packet processing applications can be enabled on a KNOX platform. Network traffic optimization, split billing and network access control are a few examples of applications that integrate with the KNOX VPN framework. The KNOX VPN framework ensures proper chaining of the traffic when multiple applications are configured to process network packets.

System, Per-App, and Workspace VPN

MDM vendors can configure the following kinds of VPN profiles:

- A System VPN profile affects all traffic from the device. There can only be one System VPN profile in effect at a time. However, per-app VPN and Workspace VPN profiles can coexist with a System VPN profile.

- Workspace VPN profiles affect network traffic for a given KNOX Workspace. Applications outside the Workspace are not included in the VPN tunnel.

- Per-App VPN profiles affect one or more named applications. All network traffic for the configured applications is tunneled through the VPN connection.

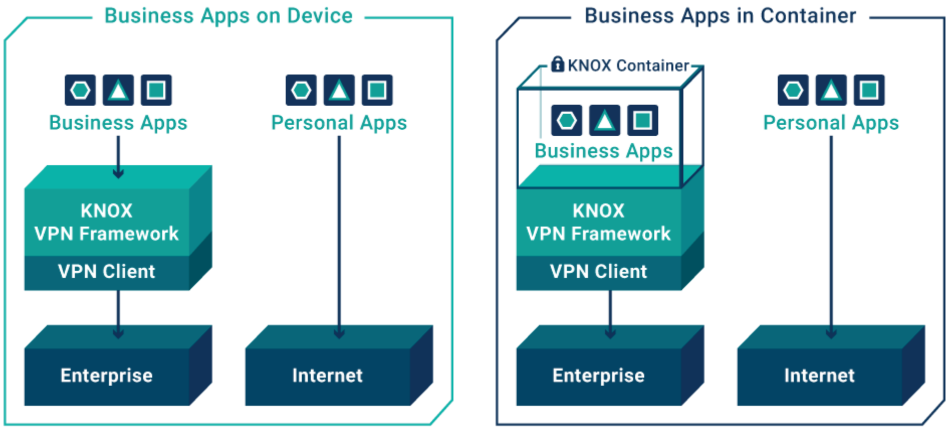


Figure 7 - Per-app VPN

For any of the VPN profiles described, all traffic for a given application is tunneled through the VPN, if it is included in the profile, irrespective of the destination address of the network packet. For example, if the browser is included in one of the VPN profiles, all browser traffic goes through VPN, whether or not the destination URL is inside the corporate network or not.

When a System VPN profile is activated along with a per-app VPN or Workspace VPN, the MDM administrator can determine the behavior as below:

- Per-App VPN or Workspace VPN connections are tunneled through the System VPN (for example chaining), or,
- Applications that are not already included in other active VPN profiles are included in the System VPN tunnel.

VPN Auto Reconnect and Always On

The Auto Reconnect feature allows failed VPN connections to be automatically restored depending on the failure condition.
Some of the key features of Auto Reconnect design are:

- When a VPN connection fails because of network/timeout reasons, connections are automatically retried.
- The Auto Reconnect feature can be enabled/disabled by the MDM administrator.
- A user can be given the option to manually retry the failed connection.

It's also possible to configure the VPN to be Always On. In this case, the VPN tunnel is automatically established after the phone boots, and remains on. Configured applications are tunneled through the VPN (whether System, Per-App, or Workspace).

SmartCard framework

The United States Department of Defense (US DoD) has mandated the use of Public Key Infrastructure (PKI) certificates for employees to digitally sign documents, encrypt and decrypt e-mail messages, and establish secure online network connections. These certificates are typically stored on a SmartCard called the Common Access Card (CAC).

The Samsung KNOX platform provides applications access to the hardware certificates on the CAC via standards-based Public Key Cryptography Standards (PKCS) APIs. This access process enables the use of the CAC card by the browser, e-mail application, and VPN client, as well as other custom government applications.

Other enterprises have growing interest to use SmartCards for the same purpose, especially those that require high levels of security and information protection.

The KNOX platform provides improved SmartCard compatibility via a software framework that allows third-party SmartCard and reader providers to install their solutions into the framework.

Single Sign-On

Single Sign-On (SSO) is a feature that provides common access control to several related, but independent, software systems. The user logs in once and has access to all systems without being prompted to log in again. For example, SSO allows access to the Workspace container (and participating apps that require credentials within the container) with one password.

SSO shares centralized authentication servers that all other applications and systems use for authentication purposes. It combines this with techniques to ensure that users do not have to actively enter their credentials more than once. SSO reduces the number of user names and passwords a user must remember, and reduces IT costs with fewer help desk calls about login credentials.

KNOX Identity and Access Management (IAM) provides a comprehensive and flexible SSO solution to support enterprise applications on Samsung mobile devices. As a pioneer of enterprise identity management, KNOX offers the first enterprise federation SSO framework for Android, the KNOX generic SSO framework.

This framework was created to reduce the complexity for enterprise applications to support SSO on mobile devices. There are many Identity Providers with different SSO solutions and with various protocol support such as SAML, OAuth, OpenID, etc. They each distribute their Software Development Kits (SDKs) to mobile app developers or Independent Software Vendors (ISVs). The ISVs must customize multiple versions of their apps to support the different Identity Providers' SSO solutions.

The KNOX generic SSO framework is a bridge between the Identity Providers and the ISVs that allows a single version of an app to work with any Identity Provider SSO solution. The KNOX SSO solution provides unified Application Programming Interface (API) for SSO token retrieval and management, called getToken. Samsung partners with leading Identity Partners including Microsoft (Azure Active Directory), CA Technologies, and Centrify. Identity Providers plug their Android Application Package (APK) authenticators into the KNOX generic SSO framework and each authenticator works as a proxy to process SSO authentication requests and responses, thereby eliminating the need for ISVs to create multiple versions of their apps.

Active Directory Integration

KNOX now provides an option for the IT admin to choose an Active Directory password as the unlock method for KNOX containers. This has two important benefits. First, it allows IT Admins to use a one-password management policy for desktop and mobile devices. Second, the end user only needs to remember one password to access all services offered by the employer, thereby reducing employee password fatigue and improving productivity.

At the heart of this feature is the proven industry-standard Kerberos protocol. Active Directory is the most widely-deployed enterprise grade directory service that has built-in support for Kerberos. KNOX provides a set of Workspace creation parameters to configure Workspace to use the Active Directory password as the unlock method. Additionally, IT Admins can also configure Single Sign-On for services inside Workspace, along with the unlock method.

Mobile Device Management

KNOX provides hundreds of Mobile Device Management (MDM) security policies for fine-grained control of devices. The solution includes:

- Mobile Device Management (MDM)
- Mobile Application Management (MAM)
- Identity and Access Management (IAM)

The broad categories of supported MDM APIs are shown on the following page.

KNOX MDM policies are designed to lower cost and improve usability and manageability for small or medium enterprises. The full mobile and web application solution has cross-platform support for Samsung devices, other Android devices, and iOS^{TM} devices to support BYOD or COPE.

Support for cross-platform devices creates a centralized location for enterprises to manage devices. Mobile Application Management focuses on data management, as well as who has access to applications.

Identity and Access Management adds another layer of security with automated user authentication and easy access for administrators to monitor all activity. IAM reduces password errors with convenient Single Sign-On (SSO) and gives IT Admins time to focus on policy enforcement.

Enterprises can use the cloud-based policy management, an on-premise Active Directory, or a hybrid combination to separate employees and external or partner users. The full mobile and web application solution has cross-platform support for Samsung devices, other Android devices, and iOS^{TM} devices to support BYOD or COPE.

KNOX MDM API Categories

Enterprise IT Compatibility

- Account Management using blacklisting/whitelisting
- Active Directory integration
- LDAP Management

Security and Compliance

- Device Admin Management
- Firewall
- Password Management
- Device Security
- Remote Event Injection
- Audit Logging
- Usability
- Kiosk Mode
- Workspace Management
- Multi-user Mode

Device Control

- Date and Time
- Bluetooth
- Location Management
- Device Restrictions
- Wi-Fi Configurations
- APN Settings
- Device Inventory

Application Management

- Browser
- Email/Exchange Configuration
- Application Management

Telephony

- Telephony Management
- SIM Change Information
- Roaming Restrictions

Simplified enrollment

Enrolling an Android device into a company's MDM system typically begins with a user downloading the agent application from the Google Play store, then configuring it for authentication. Enterprises are facing increasing help desk calls as more and more users are activating mobile devices for work. When presented with prompts, privacy policies, and license agreements, users might experience difficulties during the process, resulting in a poor overall experience.

The KNOX platform provides a simplified enrollment solution that is streamlined and intuitive and eliminates many steps and human error.

The enrollment process provides the employee with an enrollment link sent by e-mail, text message, or through the company's internal or external website. Once the link is clicked, users are prompted to enter their corporate e-mail address. This action triggers the display of all required privacy policies and agreements. After accepting the terms, users enter a corporate account password for authentication from the enterprise. Any agent application required is automatically downloaded and installed.

MDM vendors can take advantage of this feature to simplify the onboarding process for enterprise users, significantly improve the user experience, and reduce support costs.

The KNOX platform offers significant enhancements to the management policies previously offered. In particular, bulk enrollment now allows IT Admins to enroll hundreds or thousands of employees at the same time. An IT Admin portal is used to create an MDM profile where the IMEIs of employees are added. Employees receive a notification to accept enrollment, or another method can seamlessly enroll users without user authentication.

In addition, the Samsung KNOX Mobile Enrollment allows IT Admins to enroll hundreds or thousands of employees at the same time. Samsung provides a web tool and an application to scan package bar codes. The device requires end user approval before enrolling to the MDM server.

KNOX Mobile Enrollment supports multiple MDM configurations per account. With complex device environments, and multiple MDM profiles or configurations, KNOX Mobile Enrollment gives IT Admins the ability to prepare hundreds of devices and get them connected to the right MDM with ease. End users only need to turn on the device and connect to the network. KNOX Mobile Enrollment takes care of activation without users needing to do a thing.

Enterprise Billing

Enterprise Billing provides enterprises a mechanism to separate enterprise data usage from personal data usage. This enables enterprises to compensate their employees for costs generated because of work, particularly in BYOD cases, or to only pay for work-related data in COPE cases.

The KNOX platform supports Enterprise Billing from KNOX version 2.2 or above, and requires MDM support.

Enterprises configure two Access Point Name (APN) gateways. One APN is for data associated with enterprise-approved apps, and a different APN is for all other personal data. Enterprises must first register with a network operator's enterprise billing service. Once a new APN is provisioned for business use, KNOX Workspace can be enabled for that dedicated APN. IT Admins can also select individual apps inside or outside Workspace to use data over the enterprise APN.

Enterprise billing configured with a dedicated APN:

- Separates data usage over the mobile internet for 2G/3G/4G connections
- Routes all data traffic from KNOX Workspace over the enterprise APN
- Provides the capability to select individual apps inside or outside KNOX Workspace to use data over the enterprise APN

The enterprise APN can also be configured to allow or not allow roaming. When roaming is enabled, personal data is routed through the default APN, and enterprise data is routed through a dedicated enterprise APN. By default, roaming over the enterprise APN is disabled. When a user is roaming in a single Packet Data Protocol (PDP) network, all enterprise apps are automatically routed to the personal APN for work continuity.

If enterprise apps use a VPN connection to the network, the VPN profile can be configured to route data through the enterprise APN.

Dual SIM devices can also be enabled for KNOX Enterprise Billing. The primary, or first SIM slot, is automatically selected to configure an APN and activate Enterprise Billing on the device.

To avoid personal use of a SIM card, IT Admins can lock the SIM card with a unique PIN combination. This ensures that the SIM can only be used for enterprise billing on the authorized device. In addition, dedicated enterprise APNs are restricted, and APN settings are not visible or editable on the device.

Users can check personal and enterprise data usage on a KNOX device in the Settings menu. To view data usage, employees can go to Settings > Data Usage > Mobile Tab (personal) or Enterprise Tab (work).

Endnotes

[1] Juniper Networks, "Juniper Networks Third Annual Mobile Threats Report, March 2012 through March 2013," p. 4-6. http://www.juniper.net/us/en/local/pdf/additional-resources/jnpr-2012-mobile-threats-report.pdf

[2] Aspect Security, Inc., "2013 Global Application Security Risk Report," p. 2. http://cdn2.hubspot.net/hub/315719/file-681702349-pdf/presentations/Aspect-2013-Global-AppSec-Risk-Report.pdf

[3] Nielsen, "The Digital Consumer," October 2013, p. 8. http://www.nielsen.com/content/dam/corporate/us/en/reports-downloads/2014%20Reports/the-digital-consumer-report-feb-2014.pdf

[4]Consumer Reports, "Smart phone thefts rose to 3.1 million last year, Consumer Reports finds," May 2014. http://www.consumerreports.org/cro/news/2014/04/smart-phone-thefts-rose-to-3-1-million-last-year/index.htm

[5] FCC, "Report of Technological Advisory Council (TAC) Subcommittee on Mobile Device Theft Prevention (MDTP)," December 2014, p. 22. http://transition.fcc.gov/bureaus/oet/tac/tacdocs/meeting12414/TAC-MDTP-Report-v1.0-FINAL-TAC-version.pdf

[6] Workshare, "Data Guardian: Detecting Business Risk 2014," p. 14-16. https://d3liiczouvobl1.cloudfront.net/uploads/refinery/resource/file_name/251/Workshare_-_Data_Guardian_-_Detecting_Business_Risk_2014.pdf

About Samsung Electronics Co., Ltd.

Samsung Electronics Co., Ltd. is a global leader in technology, opening new possibilities for people everywhere. Through relentless innovation and discovery, we are transforming the worlds of televisions, smartphones, personal computers, printers, cameras, home appliances, LTE systems, medical devices, semiconductors and LED solutions. We employ 236,000 people across 79 countries with annual sales exceeding KRW 201 trillion. To discover more, please visit www.samsung.com

For more information about Samsung KNOX, visit www.samsung.com/knox

Samsung Electronics Co., Ltd.
416, Maetan 3-dong, Yeongtong-gu
Suwon-si, Gyeonggi-do 443-772, Korea

Version	Date
Samsung KNOX Security Solution V1.05	April 24. 2015
Samsung KNOX Security Solution V1.04	April 13, 2015
Samsung KNOX Security Solution V1.03	April 10. 2015
Samsung KNOX Security Solution V1.02	March 18, 2015

EXHIBIT 2

SAMSUNG
Knox

Whitepaper:

Samsung Knox™ Security Solution

Version 2.2 May, 2017
Samsung Research America
Samsung Electronics Co., Ltd.

Contents

Section 1: BYOD and mobile security 4

Section 2: Background: What's in a smartphone? 5

Smartphone hardware 5

The Android operating system 6

Mobile device security 7

Section 3: Samsung Knox overview 9

The Samsung Knox philosophy 9

Samsung Knox design 10

Problem: Lack of trust in Android security 13

Solution: Base security in a hardware-rooted trusted environment 13

Problem: Mixing enterprise data and user apps on one device 16

Solution: Protect enterprise apps and data in a secure Workspace 16

Problem: Device theft 17

Solution: Protect enterprise data-at-rest by default 17

Problem: Difficulty of securely implementing custom enterprise applications 18

Solution: Provide Knox security services to enterprise applications 19

Problem: Lack of enterprise manageability and utilities 19

Solution: Provide extensive manageability and utilities 19

Certifications 21

Section 4: Technology in depth 23

Part 1. Building a hardware-rooted trusted environment 23

Part 2. Making the trusted environment enterprise ready 37

Section 5: Enterprise readiness 43

Endnotes 56

Section 1: BYOD and mobile security

Personal smartphones and mobile devices permit employee access to corporate email and network resources, but add a vulnerable entry point that can be exploited. Additionally, document sharing enables mobile devices to further proliferate corporate resources outside of a potentially compromised network infrastructure. The evolution of *Bring-Your-Own-Device* (BYOD) and *Corporate-Owned-Personally-Enabled* (COPE) strategies started slowly, and then accelerated with the proliferation of apps for every business and personal need. While enterprise employees enjoyed the freedom and productivity of continuous connectivity, IT admins on the other hand were blind-sided with protecting corporate owned devices from the massive amounts of insecure personal data employees began keeping on their phones.

Legacy enterprise IT admin security models were designed to protect the integrity of the enterprise network and company-issued PCs, not the personal smartphones and tablets utilized by the enterprise's employees. With both BYOD/COPE and cyber-attacks increasing, the scramble to analyze the facts and figures ensued in hopes of finding a way to manage the escalating problem and complexity of mobile device security.

What are the numbers? What are the risks?

Mobile malware attacks increased 300% between 2015 and 2016 according to a new 2017 issued report from Kaspersky labs

In 2016, the number of reported malware cyber-attacks was more than 8.5 million; that's three times more than reported in 2015, according to a Kaspersky Lab report on mobile device malware growth.

Additionally, Kaspersky registered almost 40 million attacks by malicious mobile software over the course of the 2016 as well.[1]

IT admins must quickly identify potential threats and understand how the threat profile is growing, since mobile malware attacks increased more than 300% between 2015 and 2016.[2]

Malicious and poorly designed mobile device applications aren't the only security threat in the mobile landscape, but they are the biggest threat. A Nielsen February 2014 report, The Digital Consumer, reported that smartphone owners spend 86% of their time using apps versus the mobile web.[3] Consequently, the real culprit for poor mobile security is the open source code hackers can easily use to create and distribute malicious apps.

Even while these figures were being reported, Samsung was already at work designing a solution for mobile devices. In 2012, a group of Samsung engineers built a new mobile environment, deeply rooted in the hardware of the Android *operating system* (OS) that could be used for any other mobile OS The blueprint for this solution included maintaining a highly trusted platform, and the tools required for an enterprise-ready security solution. In 2013, Samsung released Samsung Knox (TM) for any size enterprise, affording them complete control over how they implement their mobile security model.

Section 2: Background: What’s in a smartphone?

There is much more to a smartphone than the apps and widgets users typically experience. Behind the mobile device interface exists a sophisticated system of advanced processor architectures, operating system kernel, libraries, middleware, and security services.

Smartphone hardware

The *processor* is a smartphone's central computational unit where its apps and OS reside and run. The processor is physically connected to the phone’s antennas, internal storage drives, removable SD cards, and docking ports.

The modes of execution represent one of the most important features of a processor. A *mode* defines how much privilege a piece of software running on the processor has been granted. For example, when user-installed apps run on the processor, the processor runs in *user mode*. In user mode, the apps are not allowed to directly access hardware devices or resources controlled by other apps. On the other hand, when critical operating system software is running, the processor is in *privileged mode*. In privileged mode, the system’s software can directly access hardware devices, as well as all data held by the user’s applications. Any code running in privileged mode must be protected from control by adversaries wishing to exploit the device.

Knox leverages a processor architecture known as ARM® TrustZone® . While TrustZone maintains the two modes described above, it also provides a new security-specific construct called *worlds*. In TrustZone, there are two worlds, the *Normal World*, and the *Secure World*. Virtually all smartphone software as we know it today still runs in the Normal World. The Secure World is reserved for highly-sensitive computations, such as those involving *cryptographic keys*. Knox utilizes the TrustZone's Secure World extensively for protecting enterprise confidential data and monitoring the OS kernel running in the Normal World.

The Android operating system

A smartphone's hardware processor utilizes both a user and privileged mode for various functions. The portion running in privileged mode is called the *kernel.* OS kernels are among the most rigorously engineered pieces of software in the world, since they must perform many functions, all with the power of the processor's privileged mode. For example, any time phone data arrives from the Internet, the OS kernel first chooses whether to even allow the data to proceed, or to drop it. If the data is allowed, the kernel examines it and decides which application the data is intended. The kernel then places the data in the app's memory, and notifies the app data has arrived. If the app wishes to send a reply, the app's reply is sent by repeating this whole process in reverse.

Given this example, consider what could happen if an attacker gained control of the OS kernel. Due to the kernel's high permissions, the attacker could leak arbitrary sensitive data from any application, and send it anywhere on the Internet. This is a compelling reason why Knox implements its extensive protections for OS kernels.

Inter-application processing

When applications communicate with one another they ask the kernel to establish lines of communication. To facilitate more robust app communication, the Android OS provides another layer of software, called *middleware*.

The Android middleware runs in user mode. The middleware provides rich methods that allow apps to share their data and perform operations on each other's behalf. For example, many image library apps can take photos, even though they don't know how to use the phone's camera. Instead, they simply request that an app that does understand the camera take the picture on their behalf. A second major function is ensuring such communication does not occur between enterprise apps and user apps. This prevents sensitive data from leaking to an untrusted third-party, and prevents corrupted data from entering enterprise apps.

Boot process

The boot process binds the hardware, kernel, and apps. When a device is initially turned on, the user's applications are not immediately available. Instead, a succession of software components start, with each component starting the next one in the chain. Typically, when the user presses the ON button, the device first runs a program called a *bootloader*. Many mobile device architectures use multiple bootloaders to perform different functions. The bootloader then finds where the kernel is stored, and begins running the kernel in the processor's privileged mode. The kernel starts the Android middleware and some basic apps, running them in user mode. Once the boot completes, the user is queued to login into their phone.

Mobile device security

Mobile security is quite different from other domains, in that device owners have complete control over how to use and secure their own devices, as opposed to a corporate-owned laptop, controlled by IT admins. With COPE and *Corporately Owned, Business Only* (COBO), sensitive emails are likely downloaded to the device, but the user may simultaneously compromise the kernel's security to allow for device customizations. The process of unlocking a device's operating system so you can install unapproved apps is typically known as device *rooting*.

The security breach inherent in device rooting makes it easier for malicious parties to exploit the rooted device. The same techniques are known by malware authors, and can enable them to steal the user's data or attack different targets. In response, many of the security measures implemented by Knox are designed to either prevent rooting, or mitigate the resulting damage.

Cryptography

Cryptography is another fundamental measure for mobile device security. Knox uses cryptography for three key functions:

- *Encryption* - the random re-arrangement of data using a protected key
- *Hashing* - the creation of a unique series of numbers to represent a particular piece of software or data; a single difference in a piece of data yields a different hash
- *Signing* - the encryption of hashed data using a private key to prove the data originated from a particular known entity

Knox frequently uses signing to produce the hash signatures of firmware components. This proves the firmware component originated from the owner of the private key used for signing, and proves the component originated from Samsung. Knox signing keys are only accessible in the TrustZone Secure World.

Section 3: Samsung Knox overview

Enterprise data is finding its way on employee smartphones as the new norm for corporate efficiency. Corporate smartphones have increased productivity by placing work and personal data on the same device, but has compounded enterprise security exponentially. Mobile devices expose numerous pathways through which sensitive data can be stolen (sharing data with untrusted third-party applications, device theft, intentional rooting, misconfigured, or vulnerable enterprise applications). These vulnerabilities grow exponentially across an organization, when hundreds or even thousands of devices require secure management, configuration, and deployment.

Knox is the most comprehensively secure and manageable mobile device solution for enterprises large and small. Based on the Android OS, Samsung Knox is designed on the philosophy that device security should be rooted in fixed hardware mechanisms. Knox bases this foundation in the principles of *trusted computing*, a set of methods for making devices that can prove to enterprises they are running the correct security software, and can raise alerts when tampered. On top of this more stable foundation, Knox builds a workspace environment to protect enterprise apps and their data, a robust set of data at rest protections, and a large suite of enterprise security tools, including a highly configurable *Virtual Private Network* (VPN), *Single Sign On* (SSO) and *Enterprise Mobility Management* (EMM) interfaces.

The Samsung Knox philosophy

Samsung designed Knox using a industry leading two-step design philosophy:

Step 1.Build a trusted environment rooted in proven hardware security mechanisms.

In a trusted environment, sensitive or enterprise-critical functionality is only enabled once the device is in an allowed state. In this context, *state* refers to the security-relevant software and configurations on the device. For example, parts of state considered by Knox include the bootloaders, kernel, TrustZone, and numerous security policy configurations. Furthermore, the trusted environment allows for *remote attestation*, a process where any change is securely validated by a set of proofs. Third parties can then inspect these proofs to decide if the device state meets their security requirements. A device's trusted environment is rooted in hardware when its security is based on the prevention of physical tampering. Why is it important to root trust in device hardware? Knox designers are aware operating system security has historically been trusted to privileged system software (mainly the OS kernel). However, in the last two decades, attackers have become more successful at exploiting kernel flaws. Other mechanisms such as heavyweight virtual machines or special *Basic Input/Output System* (BIOS) checks have been implemented and circumvented. The Knox design recognizes the single best defense against a full-system compromise is to tie system self-checks to a secret password maintained by secure hardware and out of the reach of any software-based or physically present adversary.

Step 2. Make the trusted platform ready for enterprise use

Enterprises require a trusted platform for their critical data security. Such accessibility involves giving enterprises complete control over their data. Knox includes a collection of useful secure applications and utilities that enable enterprise-ready deployment. Knox Workspace security is based in the hardware root of trust and isolating the workspace from the personal space.

Samsung Knox design

Knox addresses the most pressing security problems facing enterprises' COPE and COBO strategies today. Samsung has identified the following key challenges in making an Android-based system enterprise ready:

- Device rooting
- The mixing of enterprise data with user apps on the same device
- Device theft
- The difficulty of securing custom enterprise applications
- The lack of enterprise manageability and supporting utilities

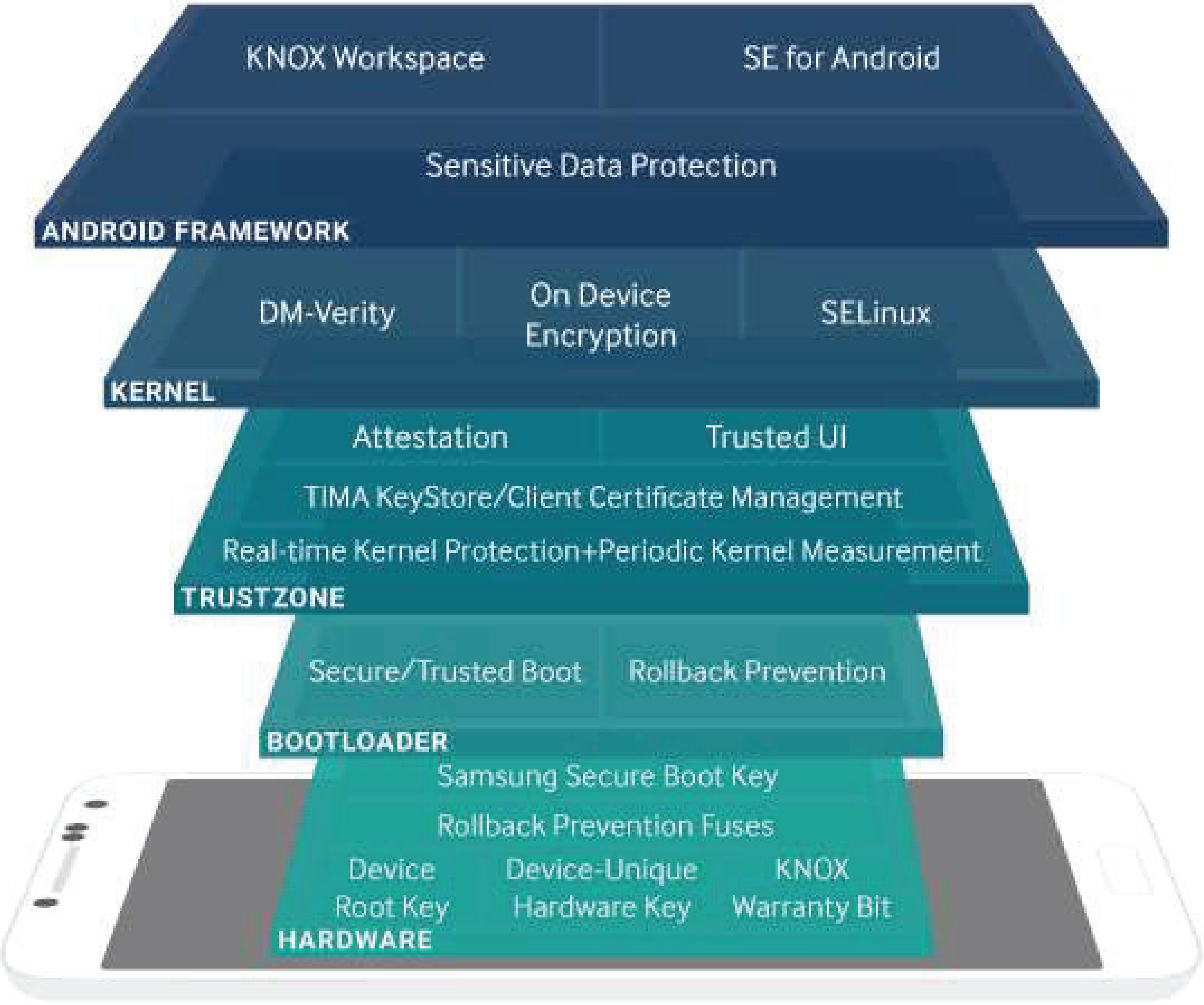


Figure 1 – Samsung Knox Security Solution

Table 1 summarizes the security challenges to an enterprise's Android adoption utilizing a Knox-specific solution.

Knox Strategy	Problem(s) Solved	Solution Technologies
Build a Hardware-Rooted Trusted Environment	Lack of trust in Android security	Hardware Root of Trust Samsung Secure Boot Key, Rollback Prevention Fuses, Knox Warranty Bit, *Device Root Key* (DRK) Build Trust Trusted Boot using *TrustZone-Based Integrity Measurement Architecture* (TIMA), Rollback Prevention Maintain Trust *Real-Time Kernel Protection* (RKP), *Periodic Kernel Measurement* (PKM), DM-Verity Prove Trust TIMA Attestation
Make Trusted Environment Enterprise-Ready	Mixing enterprise and user apps on one device	Knox Workspace, Security Enhancements for Android
	Device theft exposing enterprise data	Knox Workspace Encryption, *Sensitive Data Protection* (SDP), *On-Device Encryption* (ODE)
	Difficulty of securely implementing custom enterprise applications	*TIMA KeyStore, Client Certificate Manager* (CCM), *SE for Android Management Service* (SEAMS)
	Lack of enterprise manageability and utilities	*Enterprise Mobility Management* (EMM), *Virtual Private Network* (VPN), Active Directory Integration

Table 1 - Knox Solution Technologies

Problem: Lack of trust in Android security

The Android OS was originally designed for end users, not enterprises. The original Android approach to security was to simply isolate apps from interacting with one another. However, this design does not necessarily translate into enterprise confidence. For example, how can enterprises be sure security measures are enabled when users can root their device and intentionally exploit privileged software and circumvent vendor provided security measures?

Solution: Base security in a hardware-rooted trusted environment

Knox protects enterprise data by building a hardware-rooted trusted environment. A trusted environment ensures enterprise-critical operations can only occur when the device is in an allowed state. For software components such as the kernel and TrustZone apps, the allowed state is the required cryptographic signature of each piece of software. A trusted environment is hardware-rooted if both the cryptographic keys and code used to compute these signatures are tied back to unmodifiable values stored in hardware. Knox facilitates a hardware-rooted and trusted environment by:

1. Enforcing only approved versions of system-critical software be loaded
2. Ensuring system-critical software is not modified once loaded
3. Proving only approved system-critical software is loaded and run on a particular device when requested by the enterprise

Figure 2 on the next page shows how Knox builds, maintains, and attests its trusted environment.

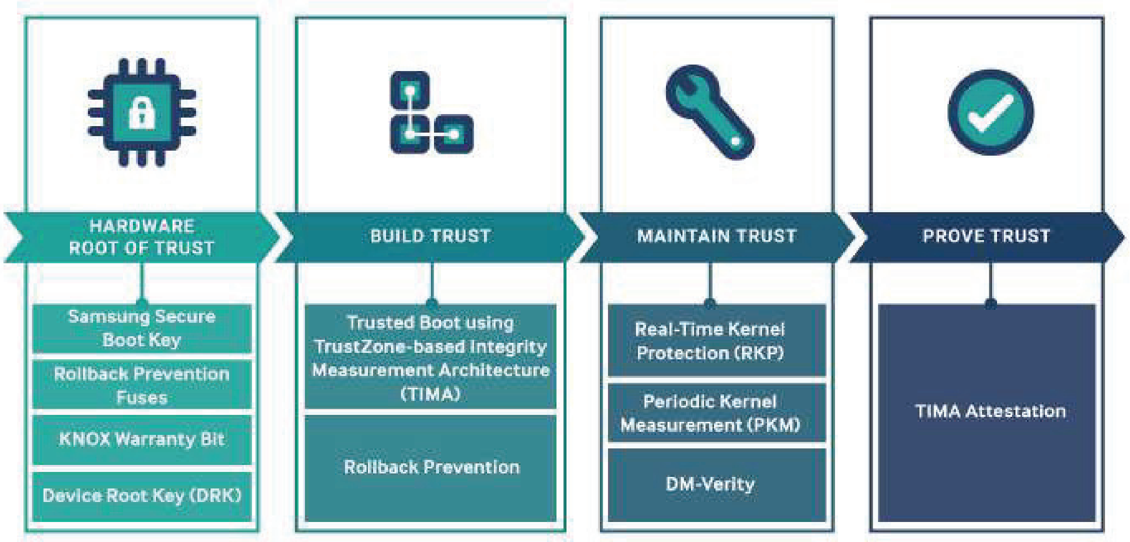

HARDWARE ROOT OF TRUST
Samsung Secure Boot Key
Rollback Prevention Fuses
KNOX Warranty Bit
Device Root Key (DRK)
BUILD TRUST
Trusted Boot using TrustZone-based Integrity Measurement Architecture (TIMA)
Rollback Prevention
MAINTAIN TRUST
Real-Time Kernel Protection (RKP)
Periodic Kernel Measurement (PKM)
DM-Verity
PROVE TRUST
TIMA Attestation

Figure 2 - Knox Builds, Maintains and Attests Its Trusted Environment

Building trust with secure and trusted boot

User applications are not immediately available when a device is powered on. Instead, a succession of software components initiate, with each component starting the next in the chain. Once initially powered, the hardware starts a bootloader running the operating system kernel. The kernel is a highly privileged component that starts applications and accesses storage and network devices directly.

Knox devices use *Secure Boot* to establish a successive component boot and signature recognition chain to verify the integrity of the each component. The device boot process halts if the signature verification process fails.

Secure Boot is limited, since it cannot distinguish between different approved versions, for example, a bootloader with a known vulnerability and a later patched version, as both versions have valid signatures. To address this limitation, Knox adopts Trusted Boot in addition to Secure Boot. With Trusted Boot, each software component in the chain measures and securely stores the cryptographic hash of the next component in TrustZone Secure World memory before loading it.

Storing and archiving measurement data enables third-parties to utilize *attestation* to identify the exact software version currently running on the device. This method ensures only the latest patched software versions are utilized to ensure patched software is not downgraded to an insecure version.

If signature verification fails and tampering is detected, Knox either blows a one-time fuse, called the Knox *warranty fuse*, or prevents further booting, depending on the configuration.

Both Secure Boot and Trusted Boot have their trust rooted in hardware. The first piece of software loaded is the primary bootloader, which resides in hardware-protected *Read-Only Memory* (ROM). The Samsung Secure Boot Key is the cryptographic key used to verify signatures, and is also stored in the device's hardware fuses.

Maintaining trust with runtime protection

Only Knox approved system software versions (such as the TrustZone OS) load at the end of a successful secure boot. Once loaded, they can then be modified. For example, users may either intentionally or unintentionally run code that exploits a flaw to maliciously modify the kernel, thus rendering it compromised. Knox detects kernel compromises quickly using *Real-Time Kernel Protection* (RKP) to actively prevent kernel code modification, and *Periodic Kernel Measurements* (PKM) that periodically check kernel code integrity. RKP checks occur in an isolated environment inaccessible to the kernel, so potential kernel exploitation cannot be extended to compromise RKP. Depending on the device model, this isolated environment can be either the TrustZone Secure World or ARM virtualization extensions. The ARM architecture virtualization extensions enable the implementation of an isolated virtual machine *hypervisor* to securely isolate RKP from the Android OS kernel. Both environments are hardware-protected and isolated from the Normal World. PKM checks occur within the TrustZone's Secure World.

The kernel is not the only attractive target for malware and malicious users. There are large numbers of other code objects and configurations that can be used by malware to become persistent, and restart each time the device restarts. Knox prevents such modifications by integrating Google's DM-Verity, a kernel module that verifies the integrity of applications and data stored on the critical system partition. If a malicious process or system partition modification is detected, DM-Verity flags the modification the next time the data is read, and blocks any attempt to access the modified data.

Proving trust

Consider an EMM server that wishes to interact with a mobile device. The EMM should not simply assume a device is not compromised. Instead, a Knox-enabled device provides the EMM with an *attestation*, a cryptographically verifiable collection of device state measurements. This attesation includes bootloader hashes, kernel, TrustZone, and logs from runtime protection mechanisms, among others. The EMM can then approve or reject the items on the list. Attestation is signed using a key derived from a hardware protected *Device Root Key* (DRK). A trusted environment is proved to a third party when the device hardware is validated as not tampered or exploited, and the ARM TrustZone Secure World software works properly. Knox solves the security threats described in subsequent sections of this whitepaper via technologies Samsung built on top of this trusted environment. Without the trusted environment in place, all remaining security measures are ineffective, as there is no guarantee they were even loaded or will run as expected.

Problem: Mixing enterprise data and user apps on one device

One of the major BYOD and COPE challenges facing enterprises is the convergence and interaction of sensitive enterprise apps and data with potentially malicious user-installed apps. Android provides apps with many ways to interact with one another. Apps may share databases containing photos or contact information, and perform actions on each other's behalf, such as opening a link in an SMS text message. This interoperability has greatly benefited the mobile ecosystem, resulting in an explosion of useful applications. However, shared data from sensitive enterprise emails and documents can easily be leaked to untrusted apps claiming to provide a parallel task. Enterprises must guarantee their data is safe, even with hundreds or thousands of employees downloading untrustworthy third-party apps.

Solution: Protect enterprise apps and data in a secure Workspace

Samsung's strong isolation utilizes *Mandatory Access Controls* (MAC) to secure enterprise and user applications on the same device. With MAC, access to resources is restricted and can only be modified by the device vendor, in this case, Samsung.

Samsung adapted the *Security Enhancements for Linux* (SELinux) MAC system for Knox to provides a rich policy language for describing fine-grained access to resources by programs. Samsung extends SELinux into *Security Enhancements for Android* (SE for Android). SE for Android provides additional mediation locations in Knox's Android middleware, along with additional policy language. Knox's pioneering of SE for Android has led to its adoption into the *Android Open Source Project* (AOSP). The Knox *Workspace* is built on top of SE for Android to define a protected environment for apps and data. The workspace environment includes the home screen, launcher, applications, and widgets. The workspace functions alongside the user's environment, but it is protected from interference from user-installed applications. Data created by containerized applications are kept on a protected partition. Tampering detected during Trusted Boot execution renders the workspace and its data inaccessible.

Problem: Device theft

Smartphone theft is one of the most serious threats to confidential enterprise data.

A 2016 Consumer Reports study noted, “Smart phone thefts rose to 3.1 million last year,” and the report estimated that only 7% of smartphone users enable encryption for data at rest. Furthermore, only 36% enable a screen lock, and 34% take no security precautions at all. [5] Also, a recent FCC study estimates nearly 10% of all thefts and robberies in the US in 2013 were related to mobile device theft or compromised user data.[4] Consequently, a secure mobile OS must protect data without a user having to periodically invoke periodic measures.

Solution: Protect enterprise data-at-rest by default

Samsung Knox doesn't depend on device users to secure their own BYOD or COPE devices. Knox defines two data classes – *protected* and *sensitive*. All data written by apps in the secure workspace is considered Knox protected. Data is encrypted on disk when the device is powered off. In addition, the decryption key for protected data is tied to the device hardware. This makes protected data recoverable only on the same device. Access controls are used to prevent applications outside the Knox Workspace from attempting to access protected data.

Even stronger protection is applied to sensitive data. Sensitive data remains encrypted as long as the workspace is locked, even if the device is powered on. When the user unlocks their Knox Workspace using their password, *Sensitive Data Protection* (SDP) allows sensitive data to be decrypted. SDP keys are cleared when the user re-locks the workspace. The SDP data decryption key is tied to both device hardware and user input. Therefore, the data is recoverable only on the same device and with user input.

SDP can be used in one of two ways. First, all emails received are considered sensitive, and are immediately protected by SDP encryption. Emails received when the workspace is locked, are immediately encrypted, and can only be decrypted the next time the workspace is unlocked. The second way to use SDP is through the Knox *Chamber*. The Chamber is a designated directory on the mobile device file system. Any data placed into the Chamber is automatically marked as sensitive by Knox and protected by SDP.

Problem: Difficulty of securely implementing custom enterprise applications

Properly implementing cryptography, authentication, and secure storage services proposes unique challenges. Vulnerabilities exist in cryptographic libraries and applications that leak keys. Once a key is leaked, data previously encrypted with that key becomes vulnerable. Keeping secret keys secret is a problem for a number of reasons. First, many applications make multiple copies of keys in their internal logic, which they do not properly track and delete, thus increasing the risk of key leakage. Aggravating the problem, implementation flaws can expose secret keys directly to the network. In spite of the risks associated with implementing cryptographic services in applications, many enterprise apps require them.

Solution: Provide Knox security services to enterprise applications

Knox enables enterprise developers to build custom applications on top of its proven hardware-rooted trusted environment. Knox exposes APIs for key management, and FIPS 140-2 compliant cryptographic algorithms. The Trusted Boot-based TIMA KeyStore provides applications one type of secure key storage. Recall Trusted Boot only allows sensitive operations to occur if approved versions of all security-critical system software are loaded. The TIMA KeyStore stores all application keys in the TrustZone Secure World storage. From there, the keys can only be accessed if Trusted Boot is successful. Thus, an application's keys are safe, even in the event a user or malicious application tampers with critical system software.

Client Certificate Management (CCM) is similar to TIMA KeyStore as a complementary service for generating, storing and using asymmetric key pairs and certificates in TrustZone Secure World storage. The CCM API equips applications with PKCS#11 compliant token management, and public key algorithms for signatures and encryption.

Problem: Lack of enterprise manageability and utilities

Knox affords IT admins the functionality and manageability required for enterprise optimization. First, to use the secure environment effectively, enterprises need utilities such as a VPN and Microsoft Exchange integration. Second, enterprises need control of their devices and utilities to configure the workspace to meet their security needs.

Solution: Provide extensive manageability and utilities

Samsung Knox provides enterprises the controls to configure their workspace using an extensive set of more than 1500 APIs.

Samsung Knox utilizes per-application VPN controls, and smartcard framework integration with Microsoft Active Directory. Additionally, *Knox Mobile Enrollment* (KME) and *Knox Configure* (KC) are available to IT admins to securely enroll devices in bulk to exponentially reduce deployment times.

EMMs can prove their devices are running a trusted environment using TIMA attestations. TIMA attestation data contains boot component cryptographic hashes and security information. Data is signed using a key derived from the DRK, which proves that the attestation data originated from the TrustZone Secure World.

Samsung Knox has a dedicated team to help determine an enterprises' deployment needs and prepare custom Knox flavors if you enterprise requires customizations beyond what is offered by your EMM.

Samsung Knox has obtained a number of certifications that may be helpful if your enterprise requires compliance with specific security policies:

Certifications

FIPS 140-2 Certification

Issued by the *National Institute of Standards and Technology* (NIST), the *Federal Information Processing Standard* (FIPS) is a US security standard that helps ensure companies that collect, store, transfer, share, and disseminate *sensitive but unclassified* (SBU) information and *controlled unclassified information* (CUI) can make informed decisions when choosing devices for their workplace.

Samsung Knox strictly adheres to FIPS 140-2 Level 1 certification for both *data-at-rest* (DAR) and *data-in-transit* (DIT).

DISA Approved STIG

The US *Defense Information Systems Agency* (DISA) publishes *Security Technical Implementation Guides* (STIGs) which document security policies, requirements, and criteria for compliance with DoD policy.

DISA approved the STIG for Samsung Knox 2.x.

DISA Approved Product List

DISA has approved select Knox-enabled devices to the US DoD *Approved Products List* (APL).

Note: Select Samsung Knox-enabled devices and tablets are certified under the *National Information Assurance Partnership* (NIAP) *Common Criteria* (CC) *Mobile Device Fundamental Protection Profile* (MDFPP).

Common Criteria Certification

The Common Criteria for Information Technology Security Evaluation, commonly referred to as Common Criteria, is an internationally-recognized standard for defining security objectives of information technology products and evaluating vendor compliance. A number of Governments use Common Criteria as the basis for their own certification schemes.

Select Samsung Galaxy Knox-enabled devices received Common Criteria (CC) certification. The current CC certification targets the new *Mobile Device Fundamentals Protection Profile* (MDFPP) of the *National Information Assurance Partnership* (NIAP), which addresses the security requirements of mobile devices.

Samsung Knox is approved by the US government as the first NIAP-validated consumer mobile devices to support its full range of classified information.

Certifications

CSfC

An ever increasing number of Samsung devices have been listed in the NSA/CSS's *Commercial Solutions for Classified Program* (CSfC) for approved security components.

ANSSI

Samsung Knox has obtained first-level security *Certification Sécuritaire de Premier Niveau* (CSPN) from the *Agence nationale de la sécurité des systèmes d'information* (ANSSI). The CSPN methodology and criteria is defined by ANSSI with evaluations run by ANSSI accredited testing labs.

ISCCC

Samsung Knox received the security solution certificate from the China *Information Security Certification Center* (ISCCC). Samsung worked closely with ISCCC to develop the certification process, including device requirements and security standards. By securing the critical ISCCC certification, Samsung has a stronger foothold to garner mobile device contracts with China's regulated industries, including government authorities, ministries, and finance.

CESG Approved

The *Communications and Electronic Security Group* (CESG) approved Knox-enabled Android devices for United Kingdom government use.

FICORA

Samsung Knox devices fulfill national security requirements as defined by the Finnish National Security Auditing Criteria (KATAKRI II).

ASD

Australian Signals Directorate is approved for ASD UNCLASSIFIED via MDFPP recognition.

NOTE: For the most recent Samsung Knox certifications, go to:
https://www.samsungknox.com/en/security-certifications

Section 4: Technology in depth

The Knox design philosophy consists of the following two steps:

1. Building a hardware-rooted and highly trusted environment.
2. Making the trusted environment enterprise ready.

Part 1. Building a hardware-rooted trusted environment

Knox builds a trusted environment in four ways. Knox first builds a hardware root of trust which other components rely. Second, Knox establishes trust during boot time. Third, Knox maintains trust while the device is in use. Finally, Knox proves its trustworthiness to remote parties, such as an enterprise management system.

Figure 3 displays an overview of the Knox architecture.

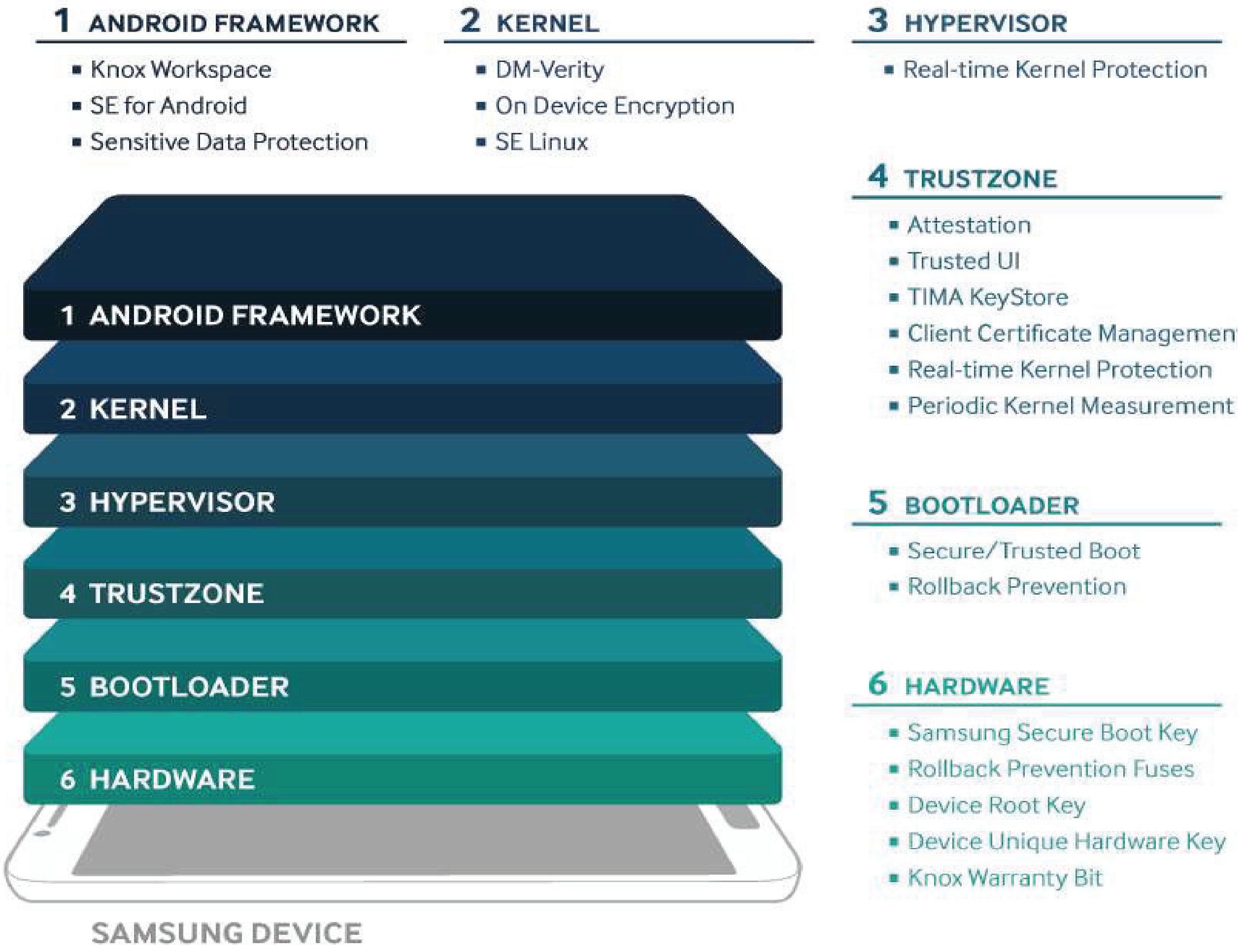


Figure 3 - Knox Architecture Overview

Hardware Roots of Trust

This section how Samsung Knox devices establish their trusted hardware environment.

Device-Unique Hardware Key (DUHK) Samsung incorporates the DUHK, a device-unique symmetric key, in the device hardware during manufacturing. The DUHK binds data to a particular device and is only accessible to a hardware cryptography module and not directly exposed to any device software. However, software can request that the DUHK encrypts and decrypts data. Data encrypted by the DUHK is bound to the device, since it cannot be decrypted on any other device.

Samsung Secure Boot Key (SSBK) The SSBK is an asymmetric key pair used to sign Samsung-approved boot executables. The private part the SSBK is used by Samsung to sign secondary and application bootloaders. The public part of the SSBK is stored in hardware one-time programmable fuses at manufacture time in the Samsung factory. The Secure Boot process uses this public key to verify whether each boot component it loads is approved.

Rollback Prevention Fuses (RP Fuses) RP fuses are hardware fuses that encode the minimum acceptable version of Samsung-approved bootloaders. Old software may contain known vulnerabilities that can be exploited. Rollback prevention prevents approved, but out-of-date versions of bootloaders from being loaded. The RP fuse version number is set when system software is initially installed and as Knox updates occur. RP fuses are programmable just once per Knox update.

Knox Warranty Fuse Knox utilizes a one-time programmable fuse that signifies whether the device has ever been booted into an unapproved state. If the Trusted Boot process detects non-approved components are used, or if certain critical security features such as SELinux are disabled, it sets the fuse. Thereafter, the device can never run Samsung Knox, access to DUHK and DRK in the TrustZone is revoked, and the enterprise's data on the device cannot be recovered.

ARM TrustZone Secure World The Secure World is a hardware-isolated environment in which highly sensitive software executes. The ARM TrustZone hardware enforces memory and devices marked secure can only be accessed in the Secure World. Most of the system as we know it, including the kernel and middleware, as well as all apps, execute in the Normal World and can never access the data used by Secure World software. The Secure World software, on the other hand, is more privileged, and can access both Secure and Normal World resources.

Bootloader ROM The *primary bootloader* (PBL) is the first piece of code to run during the boot process. The PBL is trusted to measure and verify the boot chain (see the sections on Secure Boot and TIMA Trusted Boot). To prevent tampering, the PBL is kept in secure hardware *Read Only Memory* (ROM). The device hardware loads and runs the PBL from ROM at boot, and the PBL starts the Secure and Trusted Boot processes.

Device Root Key (DRK) The DRK is a device-unique asymmetric key pair that is signed by Samsung's root key through an X.509 certificate. This certificate proves that the DRK was produced by Samsung. The DRK is generated at manufacture time in the Samsung factory and is stored on the device encrypted by the DUHK, thus binding it to the device. The DRK is only accessible from within the TrustZone Secure World.

Because the DRK is device-unique, it can be used to tie data to a device through cryptographic signatures. The DRK is not used directly to sign data; instead, signing keys are derived from the DRK. The TIMA attestation data, proving the device is in a trusted state, is signed using the Attestation Key, which is itself signed by the DRK. The DRK signature proves attestation data originated from the TrustZone Secure World on a Samsung device. Note that while the DRK is not stored directly in hardware, it is an important part of the root of trust, as it derives other signing keys, and is protected by both the DUHK and TrustZone Secure World.

Establishing trust

Android begins the startup process with the primary bootloader, which is loaded from ROM. This code performs basic system initialization and then loads another bootloader, called a secondary bootloader, from the file system into RAM and executes it. Multiple secondary bootloaders may be present, each for a specific task. The boot process is sequential, with each secondary bootloader completing its task and executing the next bootloader in the sequence, finally loading the application bootloader known as *aboot*, which loads the Android operating system. This sequence is called the boot chain.

Secure Boot

With a Secure Boot, each component in the boot chain verifies the integrity of the subsequent component against a signature before executing it. Knox stops the boot process if verification fails. Boot component signatures generate at build time using the *Samsung Secure Boot Key* (SSBK). The public part of the SSBK is stored in hardware fuses during manufacture. The first component in the chain, the primary bootloader, is stored in immutable ROM and is trusted to verify the secondary bootloader. Thus, the Secure Boot chain can only be compromised by hardware tampering. Later boot components, such as the kernel, are signed by another Secure Boot Key programmed into the previous boot component.

TrustZone-based Integrity Measurement Architecture

Secure Boot prevents a device from starting if unapproved boot components are detected. However, if the device does start, Secure Boot cannot inform a third party about what approved boot components have been loaded and run. For example, it cannot distinguish between a boot component with a known vulnerability versus a later patched version, since both versions have valid signatures. In addition, some carriers may decide to allow custom OS kernels to run on their devices. On these devices, Secure Boot cannot prevent unapproved kernels from running. This restriction poses a threat to enterprise applications and data. To remedy this Secure Boot limitation, Knox contains the *TrustZone-based Integrity Measurement Architecture* (TIMA). TIMA utilizes two features: Trusted Boot and Attestation.

TIMA Trusted Boot

In Trusted Boot, each boot component in the boot chain measures a subsequent component and stores the measurement before executing it. The Trusted Boot process flow is displayed in Figure 4, using a SHA256 cryptographic hash of the boot component. These hashes are securely stored in TrustZone-protected memory. The hash sets consist of one or more secondary bootloaders, the TrustZone Secure World operating system, the application bootloader, and the Normal World kernel. Depending on the processor make and model, additional firmware image hashes, such as the modem are included. These hashes validate device integrity to a remote server using TIMA Attestation.

Low level components tightly tied to the device hardware, such as bootloaders, should never be replaced. Any attempt to replace a low level component results in prompt informing the user to take the device to a service center for administration.

If the kernel has been modified, Trusted Boot sets the Knox warranty violation fuse. The one-time programmable memory fuse indicates the device has been tampered and cannot invoke certain Knox features thereafter. Even if the boot code is restored to its original factory state, tampering evidence remains and is reflected in the attestation results. Some device models will opt to never set the warranty violation fuse, instead always requesting the user service the device.

As bootloaders execute and takes measurements, those measurements are stored in TrustZone secure memory for future inspection using attestation.

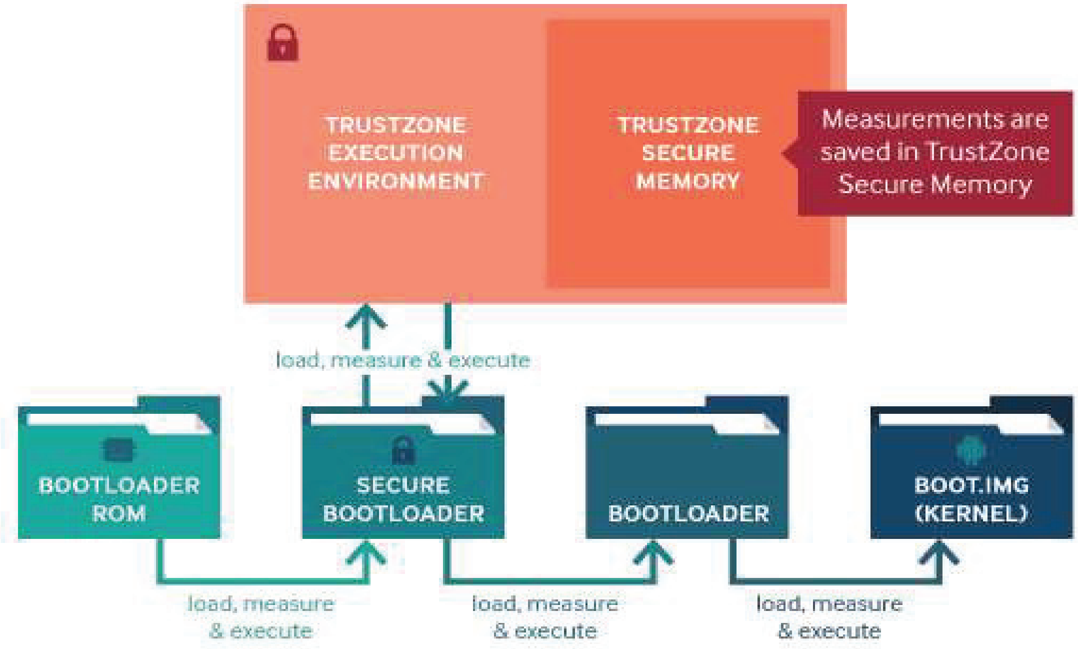


Figure 4 - The Trusted Boot Process

Rollback Prevention (RP)

Rollback Prevention blocks the device from loading or flashing an approved but old version of boot components. Old versions of software may contain known vulnerabilities that attackers can exploit. Rollback prevention checks the version of the bootloader and kernel during both boot and updates, and blocks these processes from continuing if versions are unacceptably old. The lowest acceptable version of the bootloader is stored in secure hardware fuses when the device is flashed, and the lowest acceptable version of the kernel is stored in the bootloader itself. Whenever a vendor-applied update occurs, the lowest acceptable version can be incremented in the fuses. Because this value is kept in fuses, it cannot be decremented even through physical tampering.

Maintaining trust

The following describe how Knox maintains hardware trust by protecting kernel data.

Periodic Kernel Measurement (PKM)

TIMA PKM conducts periodic monitoring of the kernel to detect if legitimate kernel code and data have been exposed to malicious software. In addition, TIMA also monitors key SE for Android data structures in OS kernel memory to detect malicious attacks could corrupt and potentially disable SE for Android.

Real-time Kernel Protection

Kernel security is essential to the Knox system. An attack that compromises the kernel has the ability to arbitrarily access system sensitive data, hide malicious activities, escalate the privilege of malicious user processes, change the system behavior, or simply take control of the system. As previously mentioned, Trusted Boot measurements determine which kernel was loaded and run when the device was started. However, this protection does not guarantee the integrity of the kernel after the system begins to interact with potential attackers. Clever attackers can often exploit an already booted and running kernel. In such cases, it is important to continuously monitor the kernel during system runtime to detect and prevent modifications to the kernel code or critical data structures.

Intuitively, the kernel protection mechanism cannot itself exist completely in the kernel, or it could be circumvented by an attacker. Therefore, Samsung Knox introduces *Real-time Kernel Protection* (RKP), a unique solution that provides the required protection using a security monitor located within an isolated execution environment. Depending on the device model, this isolated execution environment is either the Secure World of ARM TrustZone or a thin hypervisor that is protected by the hardware virtualization extensions. RKP's *Trusted Computing Base* (TCB) is part of this isolated environment and thus is secure from attacks that may potentially compromise the kernel.

Running in an isolated environment can hinder a security mechanism's ability to closely monitor events occurring inside the target kernel. To remedy this problem, RKP uses special techniques to control Normal World memory management and intercept critical events and inspect their impact before allowing them to execute. Therefore, RKP complements TIMA-PKM's periodic kernel integrity checks and their limited effectiveness against attacks and hide their traces between checks.

RKP achieves three important security initiatives:

- First, RKP prevents running unauthorized code on the system, which is accomplished by preventing the modification of kernel code, the injection of unauthorized code into the kernel, or the execution of the user space code in the privileged mode.
- Second, RKP prevents kernel data from being accessed directly by user processes. This includes the double-mapping of physical memory containing critical kernel key data into user space virtual memory. This is an important step to prevent kernel exploits that attempt to map malicious processes to kernel data regions where such regions could be modified by an attacker by an attacker.
- Third, RKP monitors some critical kernel data structures to verify that they are not exploited. In particular, RKP protects the data that defines the credentials assigned to running user processes to prevent attackers from modifying this data.

Additional Knox protection features are continually under development.

Architecture overview

Figure 5 displays the RKP architecture hosted in an isolated execution environment that's protected even if Android’s Linux kernel is compromised. The kernel pushes a request RKP to perform two operations on its behalf: (1) emulate control instructions that change the system state, and (2) update the Normal World memory translation table.

System control instructions allow the Normal World to control the security critical system state, such as defining the location of memory translation tables and exception handlers. These instructions can only be executed by privileged code, such as the kernel code. RKP works with the kernel so certain system control instructions are removed from its executable memory (the only memory-executing privileged instructions in the Normal World). Consequently, the only way to execute these instructions is by emulating them in the Secure World. Samsung calls this operation *Control Instruction Emulation*. On models using virtualization extensions, intercepting system control instructions is accomplished using hardware virtualization extensions.

Memory translation tables define the virtual-to-physical address mapping and access permissions of virtual memory. If the kernel attempts to change the current memory layout by modifying the translation tables, then RKP inspects the changes to confirm they do not impact system security. RKP ensures translation tables cannot be modified by the Normal World by making them read-only to the Normal World kernel. Therefore, the only way the kernel can update translation tables is to request the updates from RKP. As a result, RKP guarantees this security design is non-bypassable.

Kernel code protection

Kernel code protection is RKP's central feature and benefit. An attacker bypassing Linux kernel defenses is not allowed to modify the kernel executable code, significantly reducing the vulnerability of kernel attacks to the whole system. RKP examines memory translation table modifications to enforce rules so the kernel is not writable by code in the Normal World.

For detailed information on RKP and the TrustZone-based implementation of RKP, go to the ACM Digital Library website: http://dl.acm.org/citation.cfm?id=2660267.2660350&coll=DL&dl=GUIDE&CFID=629439201&CFTOKEN=91386218.

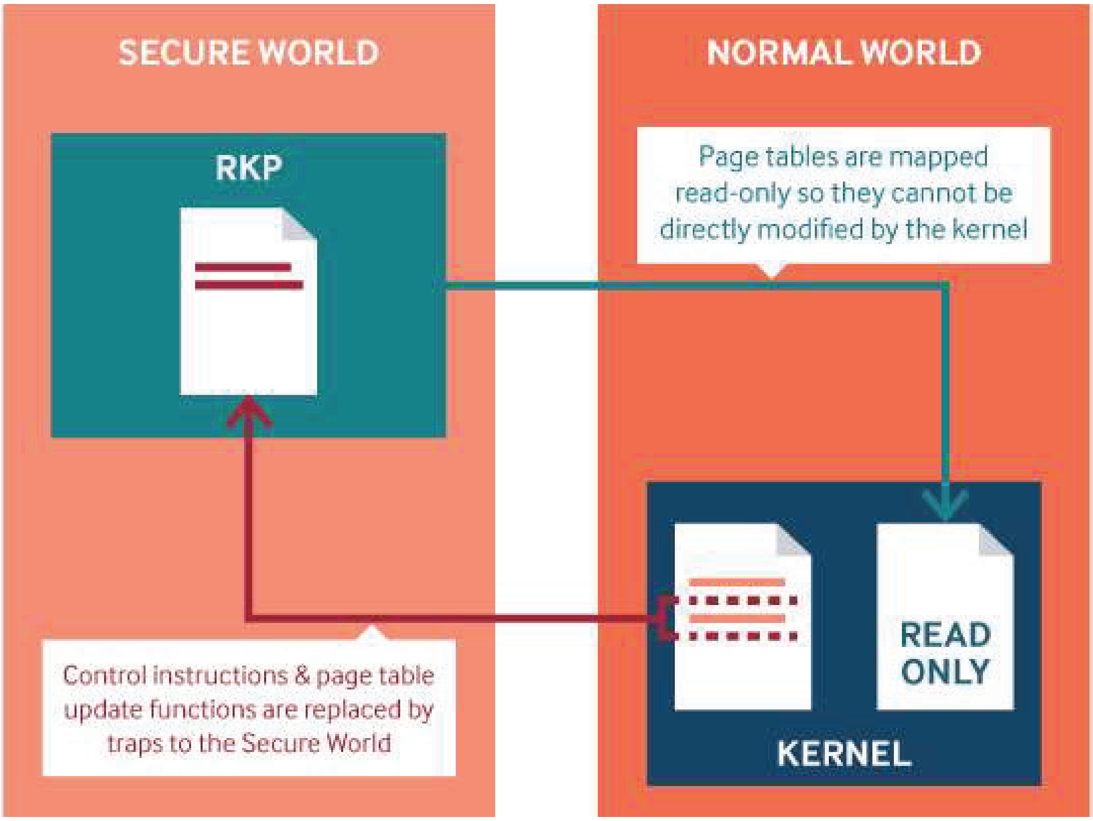

SECURE WORLD
RKP
NORMAL WORLD
Page tables are mapped read-only so they cannot be directly modified by the kernel
Control instructions & page table update functions are replaced by traps to the Secure World
READ ONLY
KERNEL

Figure 5 - RKP Architecture

RKP interoperability rules guarantee the RKP monitoring function cannot be bypassed, even when an attacker violates Normal World kernel protections. As a result, the kernel cannot modify its own code, even if compromised.

RKP rules include:

- Kernel code pages are never mapped writable un der any condition
- Kernel data pages are never mapped as executable
- Memory translation tables are mapped read-only to the Normal World
- Double mapping the kernel code or the memory translation table is not allowed. Double mapping occurs when the same physical memory is mapped to multiple virtual memory addresses, which could permit two different parts of the system to access the same memory with different permissions.
- All mapped memory regions should have the *Privileged eXecute Never* (PXN) permission, with the exception of the OS kernel.

The first two rules guarantee the initial kernel image measured by Trusted Boot, cannot be directly modified by a potential attacker unless it changes the system's memory mapping by modifying the memory translation tables. These rules remain true even if the attacker takes control of the kernel itself. The rest of the rules guarantee memory translation tables cannot be modified by the kernel, unless it sends a request to RKP. If a request is sent, RKP verifies the translation table modification does not violate the above rules. The kernel is not modified without RKP's knowledge if combining these two sets of rules together.

The kernel code protections discussed in this section assume the system memory management state has not been modified. Modifying the memory management system state (changing the effective memory translation tables' base address, or disabling virtual memory protection completely) could allow an attacker to bypass RKP monitoring. Therefore, RKP uses *Control Instruction Emulation* to inspect events to guarantee they do not tamper with its monitoring.

Knox traps system control instructions into RKP using hardware controls. In models using a TrustZone-based solution, this feature is constrained by TrustZone being incapable of trapping changes to the Normal World state. Therefore, RKP instructs the kernel to remove each system control instruction. Since these instructions can only run from privileged code, and RKP grants that privilege exclusively to the measured and protected kernel code, it is impossible for the Normal World to run these instructions without trapping to RKP. In turn, RKP validates the values written to the system control instructions to guarantee they do not invalidate its kernel code protection assumptions.

Preventing double mapping of kernel data

Kernel data structures are critical to device security. Maliciously modifying kernel data can lead to a wide range of damage. RKP uses two methods to protect the kernel codebase and prevent return-to-user attacks that exploit the operating system kernel and enable users to hijack privileged execution paths with escalated privileges. The first is through double mapping the memory hosting kernel data into the address space of the malicious process. The second is to alter the kernel control flow so it maliciously modifies its own data (such as using pointer manipulation or pointer overflow). The first attack class, double mapping kernel data to malicious user processes, is a real threat to the kernel. For instance, a real-world Android exploit used an integer overflow to trick the kernel into mapping a huge amount of the physical memory into the address space of the attacking process.

To prevent the malicious double mapping of kernel data, RKP ensures physical memory pages hosting this data are not mapped to user space processes. They can only be mapped as privileged pages and cannot be accessed by the user space. RKP enforces this rule using its control of Normal World memory translation. RKP rejects any page table modification mapping kernel data to the user space. To handle a related problem, RKP ensures no executable kernel pages are ever double-mapped as writeable, and vice versa.

RKP relies on the target kernel to inform it about the location of critical data. RKP embeds hooks in the kernel code so it is informed whenever a new memory area is allocated to the kernel. It then prevents this memory from being double mapped to writable memory anywhere else on the device.

This protection is effective against attacks using double mapping to exploit kernel data. Although RKP relies on the kernel to inform it about allocated data memory areas, this dependency does not weaken the protection. The kernel is assumed secure when it sends the information to RKP, since the data exchange occurs before the data pages are allocated. Afterwards, RKP prevents the data from being modified, except by the kernel itself.

Protecting the kernel data that defines user process credentials

The last class of attacks threatening kernel security is the alteration of the kernel control flow so it maliciously modifies its own data. These attacks may include pointer manipulation, pointer overflow, or return-oriented attacks.

Although RKP cannot fully protect against user process credential kernel attacks, it implements a novel technique to mitigate their effect by protecting selective kernel data structures critical to the system security. The data structure of choice is the *process credentials* data structure, which defines the privilege level of the user processes running inside the device. User processes represent different running applications, such as user apps. In Linux, there is an instance of the credentials structure associated with each running process. These credentials are frequently the target of rooting attacks, since a normal process can elevate its privilege through the exploitation of the credentials.

RKP uses a three-step solution to protect the credential structure from malicious modifications. First, RKP makes each instance of the credential data structure read-only by controlling the memory translation tables. Second, RKP instructs the kernel so writes to the credential structures are routed through RKP. The kernel is now unable to write to this data from within the Normal World. Before writing to the credential data, RKP examines the values to be written to make sure they do not maliciously escalate the privileges of their corresponding user process. Determining if a user process is legitimately entitled to an escalated privilege, such as the administrative privilege, is accomplished by combining multiple techniques. For example, RKP prevents processes that start with regular user privilege from escalating their privilege after they start. Additionally, processes started by applications that interface with potential attackers, such as zygote and shell, are not allowed an escalated privilege. Finally, RKP adds a check to the kernel security hooks to verify a credential structure actually belongs to the read-only memory protected by RKP before it determines the privilege of the user process. Therefore, it is guaranteed that a potential attacker cannot forge a malicious instance of the credential structures that are not monitored and verified by RKP.

For detailed information on RKP and the TrustZone-based implementation of RKP, go to the ACM Digital Library website: http://dl.acm.org/citation.cfm?id=2660267.2660350&coll=DL&dl=GUIDE&CFID=629439201&CFTOKEN=91386218.

DM-Verity

Attackers may intend to expand their exploits beyond modifying bootloader or kernel images. There are other software binaries and configuration files in storage which provide malware persistence. Persistent malware is able to restart itself each time the system is rebooted. The malware restarts by modifying programs or configurations on the system partition that contain the system binaries, Android framework, and configuration files that were started during boot. Malware can survive system reboots once inserted in the boot path. Additional problems can arise from tampering with system data and configurations, such as the granting of excessive privileges to vulnerable applications.

To prevent unauthorized modification to the system partition, Knox integrates a customized DM-Verity implementation that's a Linux/Android kernel module that performs integrity checks on all data blocks contained in a block device (such as a partition).

With stock Android, DM-Verity uses a hash tree to conduct integrity checks of individual data blocks. The hash root tree is signed by an RSA key. Whenever a data block is read into memory, DM-Verity computes the hash of the block, and then uses it, along with the other hashes on the path to the root to compute the root hash. If this computed root hash matches the signed version, the block is considered good. Otherwise, unauthorized modification of the block is detected, and the access to the data block is restricted.

Knox's DM-Verity implementation differs from stock Android in supporting file-based *firmware over-the-air* (FOTA) software updates. The Knox approach is easier to support with existing infrastructure than the stock block-based approach.

Proving trust

TIMA Attestation

TIMA Attestation enables a device to convey state information to a remote server, such as an EMM server. The attestation message contains state measurements that can be evaluated by a server, which can then decide whether to trust the device or not. A typical message contains:

- Measurements collected by Trusted Boot to demonstrate only approved system software was loaded during boot
- Security violation logs from PKM and RKP since the last reboot
- Knox warranty violation fuse status
- Device-identifying information, such as the IMEI and Wi-Fi MAC address
- A locally-computed verdict whether the device is trustworthy

The full attestation message is computed in the ARM TrustZone Secure World, and is accurate even if the Normal World OS is compromised. The verdict is a central part of the attestation message. Only when both the measurements collected by Trusted Boot match known good values, and the warranty violation fuse is intact is the verdict set to Yes to indicate attestation has passed. Good measurements are retained in a file called tima_measurement_info, maintained in TrustZone secure storage. This file is generated at build time. To simplify remote server logic, they can directly use the verdict instead of verifying all the measurements themselves.

An attestation message cannot be forged, since it's signed using the TIMA Attestation Key (and traceable to Samsung's root key). Each Samsung device supporting TIMA attestation has a unique RSA key pair, called the *Device Root Key* (DRK). The DRK is generated during manufacture and is traceable to Samsung's root key using X.509 certificates (stored in TrustZone). The remote server can verify message integrity using Samsung's root key. The signature includes a server-generated cryptographic nonce (a random number used only once) to ensure an attacker cannot replay old valid attestation messages on an already compromised device.

To illustrate this capability, consider the EMM server example previously described earlier. Depending on the attestation data and verdict, any further action is determined by the enterprise EMM security policy. The security policy might choose to detach from the device, erase the contents of the secure workspace, ask for the device's location, or any of many other potential device recovery procedure.

Part 2. Making the trusted environment enterprise ready

The next several sections describe the technologies constructed in the trusted environment to provide Knox enterprise optimizations.

SE for Android

Samsung Knox adopts the *Security Enhancement for Android* (SE for Android), which adds *Mandatory Access Control* (MAC) to *Android. Discretionary Access Control* (DAC) mechanisms, such as Android permissions or Linux owner/group/world permissions, have few security benefits since the user or process generating data has the ability to change the data's access rules. A user can make bad decisions with the data, which may then be leaked publicly. MAC provides security experts with enforcement rules that can’t be maliciously or ignorantly overridden by device users or software developers. Since these rules are mandatory, and cannot be altered, they help prevent malicious code or untrusted users from accessing sensitive data or programs. MAC can lock down data a user may want to keep secret, and prevents developers from maliciously or accidentally compromising system components that protect our devices.

SE for Android provides two layers of MAC protection:

1. Kernel-level protection: Android inherits its SELinux MAC capability directly from Linux. SELinux provides MAC for kernel system calls. A SELinux policy can enforce which objects system calls can target. For example, specify only system-signed processes can read files in the data/security directory. This level of control is possible because access check hooks are inserted inside the kernel. These hooks query the security policy before each system call to determine if it's an allowed action. SELinux policies can prevent processes from reading or tampering with data, bypassing security mechanisms, or interfering with other processes. They also reduce the damage from malicious or flawed programs

2. Android middleware protection: There's many parts of the Android system that do not leverage system calls. For example, the Android Intents used to start apps. The layer above the kernel, but below user space applications, is called the Android middleware. Additional hooks have been added to key decision points to extend MAC control to the middleware. This is known as *Middleware MAC* (MMAC). MMAC can enforce security policies among inter-component communication for Android Apps.

SE for Android security objectives include strong data and application isolation, confining the permissions of system processes running as root, and protecting applications.

Scope of access control

Samsung's custom version of SE for Android provides the following unique features:

- MAC on APIs (control who can invoke your APIs)
- Knox Workspace isolation of personal & business data
- On-the-fly workspace creation for security cusomtizations
- Quick-response policy updates (no carrier-approved firmware updates required to plug vulnerabilities)
- Strong application isolation beyond Android's access control
- Extensible MAC for new Knox features

Samsung also built an innovative global policy validation system that can detect when prohibited actions are attempted. Early detection affords Samsung unique visibility into how its devices are used and provides an earlier window to mitigate new threats before they can be exploited. This policy validation system can also refine Samsung's ability to accurately grant only the permissions needed.

SE for Android policy

SE for Android includes a set of security policy configuration files designed to meet common, general-purpose security goals. Out of the box, Samsung Knox provides a policy designed to strengthen the core Android platform and exceed enterprise needs. Samsung Knox also provides a SE for *Android Manager Service* (SEAMS), with management APIs allowing enterprise IT admins to manage SE for Android. Management tasks include gathering access logs, resetting file security labels, mapping applications to different security domains, getting type context information, and obtaining status information about packages and workspaces.

Samsung Knox provides policies to enforce the isolation of application workspaces. For example, Samsung Knox contains new security domains and can now enforce *Multiple Category Securit*y (MCS) isolation. Categories isolate applications and data into security groupings, independent of what security domain they're assigned. Categories can then ensure personal and business applications with the same security domain have their access rights limited to just their own areas. New workspaces can also be created on the fly by simply applying a new security category to a group of apps.

Sensitive Data Protection

Knox enforces two protection classes for data generated within the workspace: protected data and sensitive data. All the data generated from within the Knox Workspace is considered protected. Protected data residing in storage is always encrypted, and protected against offline attacks. Additionally, access controls prevent applications outside the workspace from attempting to access protected data. The decryption key for protected data is stored encrypted by the device-unique hardware key (DUHK). Therefore, the key is only recoverable on the same device.

Sensitive data provides an even stronger security guarantee. Like protected data, sensitive data is always encrypted when on disk. Additionally, the data remains encrypted as long as the workspace is locked. The key used to decrypt sensitive data on disk is recoverable only if the user enters the workspace password, PIN, or pattern. Thus, if a device is stolen, the key cannot be extracted from the device. Like protected data, the stored key material is encrypted by the DUHK, binding it to the device.

The enforcement of the sensitive data guarantee is conducted using Knox *Sensitive Data Protection* (SDP). SDP creates a *Container Master Key* (CMK) that can only be decrypted with user input. If desired, an EMM can also be used to unlock the CMK, preventing a total data loss in the event of a forgotten workspace password. Once the workspace is locked, SDP clears the keys in memory after a configurable timeout interval (five seconds by default). In addition, SDP also flushes sensitive file data from the OS kernel's disk cache if the file is not in use by a workspace application.

Any sensitive data received if the workspace is locked is still protected by SDP using a public key algorithm where the private part of the key is maintained in an encrypted partition, and the public part encrypts the new sensitive data. Once the workspace is unlocked, the data is decrypted with the private key, and re-encrypted using the usual symmetric key, guarded by the CMK. Currently, email subjects, bodies, and attachments are marked sensitive. Additionally, the SDP Chamber provides a designated directory on the file system. Any data placed into the Chamber is automatically marked as sensitive and protected by SDP.

On-Device Encryption

In addition to Android's kernel-level device encryption, Knox ties the encryption key to a secret maintained in trusted hardware. This is only available if the enterprise IT admin activates encryption via the EMM. TrustZone-based AES 256 *on-device encryption* (ODE) also enables enterprises to ensure device data is protected in the unlikely event the operating system is compromised. While this feature is low overhead, providing system-wide encryption means less flexibility in supporting separate security levels for user and enterprise data, thus the inclusion of the finer-grained protected and sensitive data classes.

Trusted Boot Based KeyStore (TIMA KeyStore)

The TIMA KeyStore provides applications with services for generating and maintaining cryptographic keys. The TIMA KeyStore is only enabled if the Trusted Boot measurements match the known good values in the tima_measurement_info file, and if the Knox warranty fuse is not set. Consequently, cryptographic operations with keys in the KeyStore can only occur if the system was booted into an approved state. Keys stored in the TIMA KeyStore are further encrypted with the *device-unique hardware key* (DUHK), and can only be decrypted from within TrustZone Secure World on the same device. Cryptographic operations on the keys are performed within TrustZone Secure World.

The TIMA KeyStore has the same API as familiar Android KeyStore APIs. Therefore, the only modification necessary is to specify the TIMA KeyStore used to provide the service.

Trusted Boot Based Client Certificate Management (TIMA CCM)

TIMA CCM permits the storage and retrieval of digital certificates, as well as encryption, decryption, signing, and verification in a manner similar to Smartcard functions. The certificates and associated keys are Knox encrypted with a device-unique hardware key that can only be decrypted from code running within TrustZone.

TrustZone-based CCM also provides the ability to generate a *Certificate Signing Request* (CSR) and the associated public/private key pairs to obtain a digital certificate. A default certificate is provided for applications not requiring their own certificate.

Programming interfaces for certificate storage and management are provided in the Knox Premium SDK. Application developers are provided with industry standard PKCS #11 APIs for certificate management, and therefore interact with the CCM as if it were a virtual Smartcard. Like the TIMA KeyStore, TIMA CCM operations are permitted only if the device was booted into an approved state.

Trusted UI

Knox provides a Trusted UI for secure credential entry for enterprises using PIN-based authentication. The Trusted UI uses ARM TrustZone to create a dedicated path from the device's screen and keyboard to the Secure World. Credentials entered while this path exists are completely inaccessible to Normal World programs and untrusted peripherals. Once the credentials are held in the Secure World, they are passed back to the enterprise application that initiated the authentication request.

Data erase during factory reset

Samsung's device reset procedure restores device software to its original factory default settings. The reset is completed before changing device ownership or disposing the device. Securely removing existing user data so no data is recoverable after the reset is a critical.

Erasing data on flash storage requires extra care. Samsung devices store data in a type of flash storage called *embedded multimedia cards* (eMMC). eMMC firmware uses translation tables that map device-visible logical memory to flash physical memory to improve performance and card life. This means devices cannot reference physical flash memory directly, and thus cannot ensure data is erased without support from the eMMC itself.

Samsung devices use several features supported by Samsung-manufactured eMMC chips to ensure a data erase operation during factory reset. First, when the user initiates a factory reset, the reset code instructs the eMMC firmware to discard the entire physical memory range corresponding to the logical memory storing user data. Discarded user data thereafter returns zeros when accessed by the device OS. Second, the Samsung eMMC controller firmware code responsible for discarding the physical memory is itself protected against malicious updates.

Workspace data resides encrypted in flash memory, offering yet another layer of data protection. The cryptographic keys used to encrypt Knox Workspace data are themselves stored encrypted by the device-unique hardware key, accessible only by a separate secure processor.

Section 5: Enterprise readiness

Knox Workspace: Divide and conquer

Knox Workspace is a dual persona container designed to separate, isolate, encrypt, and protect enterprise data from attackers. This work/play environment ensures work and personal data is separated, and only the work container is managed by the enterprise. Personal information like photos and messages are not managed or controlled by the IT department.

The applications and data inside workspace are isolated from applications outside workspace. Consequently, applications outside the workspace cannot use Android inter-process communication or data-sharing with applications inside the workspace. For example, photos taken with the camera inside workspace are not viewable in the Gallery outside workspace. The same restriction applies to copying and pasting. When allowed by an IT policy, some application data, such as contacts and calendar data, can be shared across the workspace boundary. The end user can choose whether to share contacts and calendar notes between the workspace and personal space. However, an IT policy ultimately controls this option.

An enterprise can manage the workspace like any other IT asset using an EMM solution; this container management process is called *Mobile Container Management* (MCM). Samsung Knox supports many of the leading solutions on the market. MCM is affected by setting policies in the same fashion as traditional EMM policies. Samsung Knox Workspace includes a rich set of policies for authentication, data security, VPN, e-mail, application blacklisting, whitelisting and so on.

Upon creation, IT admins can choose the workspace UI style (folder or launcher style), and can prevent end users from making further style changes.

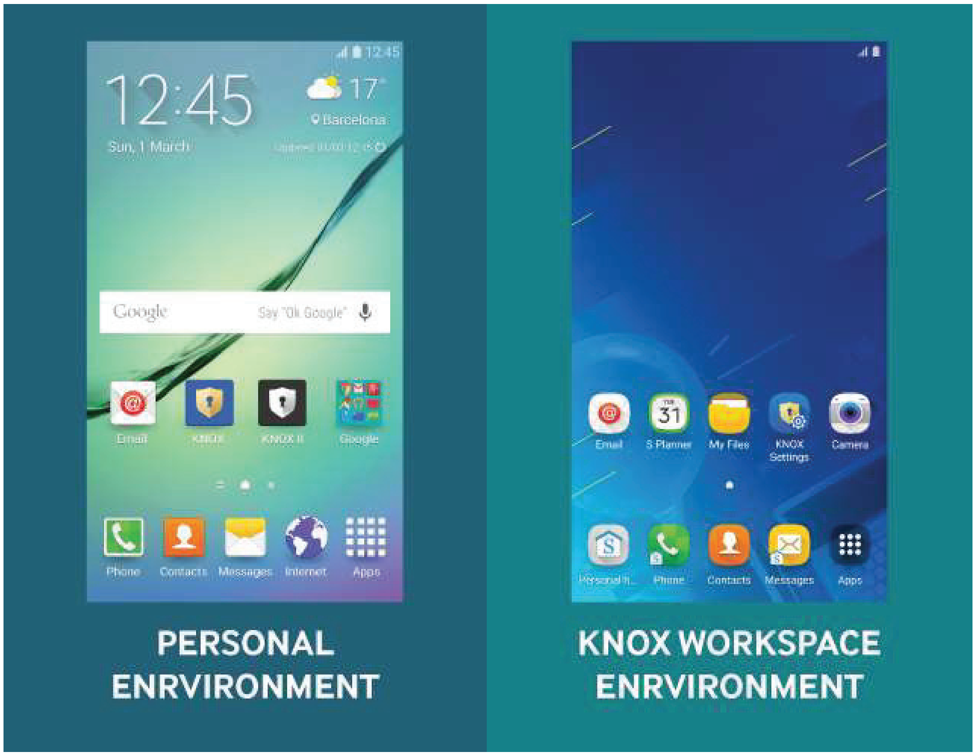


Figure 6 - User's personal environment running next to the workspace environment

Users can distinguish between the utilization and population of their personal environment applications versus workspace environment applications.

Knox Workspace also utilizes two-factor authentication. A user can set the workspace to accept a fingerprint or iris scan as the primary authenticator with a PIN, password or pattern as a second factor. The iris scan biometric authentication method is available on the Galaxy S8 platform and beyond.

The Knox platform also supports two workspaces when needed, thus meeting the needs of professionals using their own devices for corporate use and have multiple employers, such as doctors or consultants.

Other workspace features include optional Bluetooth® and *Near Field Communication* (NFC) inside the workspace itself. NFC enables a device to act as a SmartCard-based credential for physical device access and access to IT accounts. Bluetooth can be used to communicate with connected devices, and support Bluetooth profiles to enable additonal support, including printing, file sharing, and external card readers.

Applications inside the workspace can also connect with USB accessories, such as a USB printer. To propely secure the connection, IT admins must explicitly allow the USB between connectiuon between the container apps and external storage. The default for mass storage is set to OFF, and is controlled by an enterprise IT admin policy.

For Samsung Note users, a S-Pen Air Command is also supported in the workspace for writing memos, adding personal app shortcuts, screen captures, and writing notes on a screen capture (depending on the IT policy).

Knox caller ID for incoming calls, when in Personal mode, can also be configured by IT admins to display caller ID information derived from both personal contacts and Knox Workspace contacts.

Google Play for Work

IT admins can install Google Play for Work inside the Knox Workspace to silently install and uninstall apps and optionally blacklist or whitelist apps. Enterprise employees can also download IT admin approved apps in Knox Workspace. Google Play for Work can also be used outside workspace.

Additonallly, Google Voice apps inside the Knox Workspace enable users to utilize voice recognition in addition to the touchscreen keyboard.

Sensitive Data Protection (SDP) and Knox Chamber

SDP can be used in one of two ways. First, all emails received are considered sensitive, and are immediately protected by SDP encryption. Emails received when the workspace is locked, are immediately encrypted, and can only be decrypted the next time workspace is unlocked.

The second way to use SDP is through the Knox Chamber. The Chamber is a designated directory on the file system and a user-accessible folder inside the workspace. Any data placed into the Chamber is automatically marked as sensitive and protected by SDP.

Third-party application data can also be encrypted when a device is locked, then decrypted when the device is unlocked to prevent data leakage if a device is lost, stolen, or re-used. Keys required for data decryption when unlocking a device are based on the user's password.

Shared devices

Enterprises such as hospitals, banks, and airlines use shared devices for their employees. Knox supports shared devices so IT admins can manage device and security policies, and install applications with an EMM. Each employee can login separately with an Active Directory ID and password. For security and data privacy, user data is deleted when an employee logs out of their shared device.

Knox Active Protection (KAP)

If a device isn't managed by an EMM, end users can activate or deactivate *Knox Active Protection* (KAP) using the Smart Manager app. KAP uses both *Real-time Kernel Protection* (RKP) and DM Verity to provide integrity checking for system code and data. KAP is always on EMM-managed devices.

Android on a Samsung device

Android managed profiles benefit from key Knox security modules that protect the device its sensitive work data. Knox enables Android protection with the following Knox features:

- RKP actively prevents kernel code modifications
- PKM periodic kernal checks for code integrity
- DM-Verity to ensure application and data integrity on the partition
- Trusted Boot measures each software component during boot-time and securely stores the cryptographic hash of the next component in TrustZone memory before loading it
- Sensitive Data Protection APIs are available for apps in Managed Profiles. The native email app enables SDP once it's installed inside Managed Profiles.
- The TIMA and CCM TrustZone-based KeyStores provide storage for digital credentials such as VPN and email app certificates.
- Access to Managed Profiles depends on the integrity of the device. If the integrity check fails at the time of creating Android, it is not allowed. If an integrity check fails, the device cannot boot.

Android on a Samsung device does not require a Knox license activation fee. Knox security enhancements for existing Android managed profiles are updated seamlessly with *Over-the-Air* (OTA) updates.

Knox Enabled App (KEA)

Knox Enabled App is a per-app invisible container designed for application developers and vendors to provision services to device users. KEA allows service providers to deploy their applications and optimally use the Samsung Knox platform securely without the need for Enterprise Mobility Management (EMM). Since KEA is an invisible, unmanaged container, the user experience is the same as the original version of the application. Knox platform security extended to KEA provides end users data protection by encrypting app data. If a device is compromised, lost, or stolen, app data cannot be unencrypted.

The KEA workspace is implemented based on the Knox Workspace and customized according to use case requirements. Knox Workspace is created and managed by an EMM, and suitable for the enterprise environment. For individual app vendors and developers, creating, managing and configuring the KEA workspace presents challenges without an EMM. However, with KEA, the device automatically creates and manages the KEA workspace when the KEA app is installed.

Additional information (metadata) is required to operate as a KEA app. When a KEA app is installed in KEA-capable devices, the device detects the metadata and authenticates the app through a Knox License Manager (KLM) Server. Once authenticated, the KEA workspace is created and the app is installed inside the workspace, including the configuration of the *SE for Android Management Service* (SEAMS) container.

If the KEA app is installed in devices incapable of using KEA, including non-Samsung devices, the KEA metadata is ignored, and works like a regular Android app, which eliminates the need for a separate version of the app.

Virtual Private Network

The Knox platform offers additional comprehensive support for enterprise *Virtual Private Networks* (VPN). This support enables businesses to offer employees an optimized, secure path to corporate resources from their devices.

Knox offers the following VPN features for IPsec and SSL:

- Per-app connections
- On-demand connections
- Always-on connections
- Device-wide connections
- VPN chaining (nested connections)
- Blocking routes to prevent data leakage if a mandatory VPN connection drops
- Pushing VPN profiles to multiple managed devices
- Traffic usage tracking
- HTTP Proxy over VPN

Use Knox to configure VPN connections to enforce Web traffic redirection through an HTTP proxy server, allowing enterprises greater visibility into network traffic and device usage patterns of employees. The Knox VPN framework supports VPN configurations using a static proxy server IP and port, and web proxy authentication.

The Knox platform offers broad feature support for the IPSec protocol suite, including:

- Internet Key Exchange (IKE and IKEv2)
- IPsec IETF RFCs – IKEv1
- IKEv1 – Main and aggressive IKE exchange modes with pre-shared key, certificates, Hybrid RSA, and EAP-MD5 authentications
- IKEv2 with PSK and certificate-based authentication
- IKEv2 – Pre-shared key, certificates, EAP-MD5 EAP-MSCHAPv2 authentication methods, and mobile extensions
- Triple DES (56/168-bit), AES (128/256-bit) with MD5 or SHA
- IKEv1 Suite B Cryptography supported with PSK and ECDS signature-based authentications
- IKEv2 Suite B Cryptography supported with ECDSA signatures

The Knox supports leading SSL VPN vendors. Since SSL implementations are proprietary, Knox features a generic VPN framework which enables third-party SSL vendors to support their clients as plug-ins. Enterprise IT admins use Knox EMM policies to install and configure a SSL VPN client.

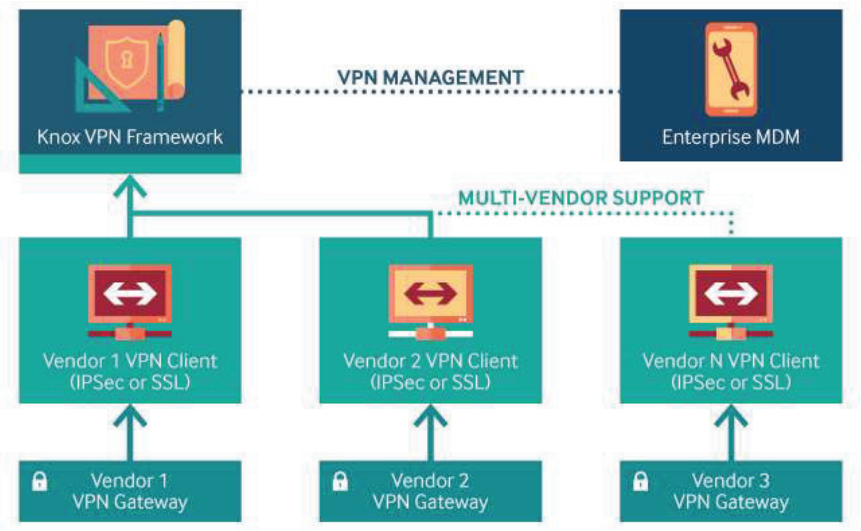


Figure 7 – Multi-Vendor Support in Knox

The per-application Knox Workspace VPN feature enables an enterprise to automatically enforce a VPN on just a specific set of applications. For example, an IT admin can configure an employee's device to enforce VPN for only business applications, ensuring data from the user's personal applications do not use the VPN and overload the company's Intranet resources. At the same time, user privacy is preserved because personal data does not enter the enterprise network.

Per-app VPN can also be applied to the Knox Workspace for some, or all, container applications.

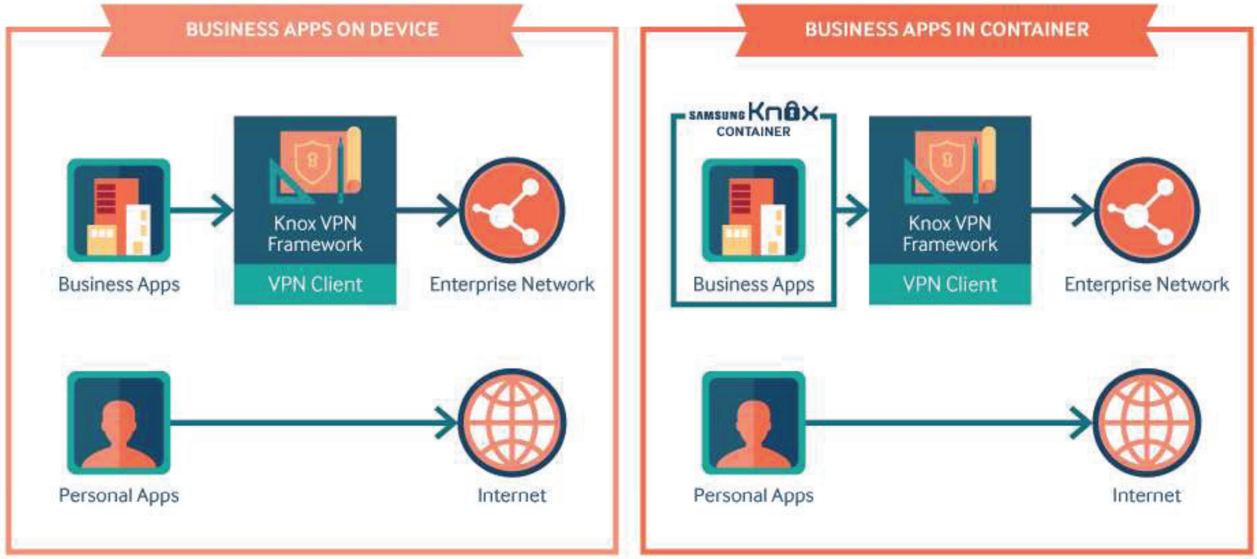


Figure 8 - Per-app VPN

Smartcard framework

The *United States Department of Defense* (US DoD) has mandated the use of Public Key Infrastructure (PKI) certificates for their employees to digitally sign documents, encrypt and decrypt e-mail messages, and utilize secure network connections. These certificates are typically stored on a Smartcard called the *Common Access Card* (CAC).

The Knox platform affords application access to the hardware certificates on the CAC via standards-based *Public Key Cryptography Standards* (PKCS) APIs. This type of access enables the use of the CAC card by the browser, e-mail application, and VPN client, as well as other custom government applications. Other enterprises show a growing interest to using Smartcards for the same purpose, especially those requiring robust security and information protection.

The Knox platform provides improved Smartcard compatibility via a software framework that allows third-party Smartcard and reader providers to install their solutions into the framework.

Active Directory integration

Knox provides an option to choose an Active Directory password as the unlock method for Knox Workspace. This has two important benefits. First, it allows IT admins to use a one-password management policy for desktop and mobile devices. Second, the end user only needs to remember one password to access all services offered by the employer, thereby reducing employee password fatigue and improving productivity.

At the heart of this feature is the proven industry-standard Kerberos protocol. Active Directory is the most widely-deployed enterprise grade directory service built-in support for Kerberos. Knox provides a set of workspace creation parameters to configure workspace to use the Active Directory password as the unlock method. Additionally, IT admins can also configure Single Sign-On for services inside workspace, along with the unlock method.

Enterprise Mobility Management

Knox provides 1,500 of Enterprise Mobility Management (EMM) security policies for fine-grained device control. The solution includes:

- *Enterprise Mobility Management* (EMM)
- *Mobile Application Management* (MAM)
- *Identity and Access Management* (IAM)

Knox EMM policies are designed to lower costs and improve device usability and manageability for small or medium sized enterprises. The full mobile and web application solution has cross-platform support for Samsung devices, other Android devices, and iOS^{TM} devices to support BYOD or COPE deployments. Support for cross-platform devices creates a centralized location for enterprises to manage devices. Mobile Application Management focuses on data management, as well as who has access to applications.

Identity and Access Management adds yet another layer of security with automated user authentication and easy access for administrators to monitor system activity.

Enterprises can use the cloud-based policy management, an on-premise Active Directory, or a hybrid combination to separate employees and external or partner users. The full mobile and web application solution has cross-platform support for Samsung devices, other Android devices, and iOS^{TM} devices to support BYOD or COPE.

Knox API categories

Enterprise IT Compatibility

- Account Management using blacklisting/whitelisting
- Active Directory integration
- LDAP Management
- Enterprise Billing
- VPN

Security and Compliance

- Device Admin Management
- Firewall
- Password Management
- Device Security
- Remote Event Injection
- Audit Logging
- Usability
- Kiosk Mode
- Workspace Management
- Multi-user Mode

Device Control

- Date and Time
- Bluetooth
- Location Management
- Device Restrictions
- Wi-Fi Configurations
- APN Settings
- Device Inventory

Application Management

- Browser
- Email/Exchange Configuration
- Application Management

Telephony

- Telephony Management
- SIM Change Information
- Roaming Restrictions

Knox Mobile Enrollment (KME)

Enrolling an Android device into a company's EMM system typically begins with a user downloading the agent application from the Google Play store, then authenticating it. Enterprises are facing escalating help desk calls as more and more users are activating mobile devices for the workplace. When presented with prompts, privacy policies, and license agreements, users might experience difficulties during the process, resulting in a poor overall experience.

The Knox platform provides a simplified enrollment solution for supported EMMs that's streamlined, intuitive, and eliminates steps and human error potential.

Enrollment occurs using either self-discovery with an email domain, or employees are provided an enrollment link sent by email, text message, or through the company's internal or external website. Once the link is clicked and invoked, users are prompted to enter their corporate email address. This action triggers the display of all required privacy policies and agreements. After accepting the terms, users enter a corporate account password for enterprise authentication. Any agent application required is automatically downloaded and installed.

KME allows IT admins to enroll hundreds or thousands of employees at the same time. Samsung provides a web tool and an application to scan package bar codes (the device IMEI). KME is targeted for devices purchased for COPE enterprises and supported carriers and resellers.

Another option includes using a master device to automatically enroll devices using NFC. The master device is configured by downloading an app from the Playstore. Each device is enrolled in an EMM profile selected by the IT admin.

EMM vendors can utilize KME to simplify the onboarding process for their enterprise users, significantly improve the user experience, and reduce support costs.

KME supports multiple EMM configurations per account. With complex device environments, and multiple EMM profiles or configurations, KME enables IT admins to prepare hundreds of devices and connected them to the right EMM with ease. End users only need to turn on the device and connect to the network. KME takes care of activation without users needing to do a thing.

Enterprise Billing

Enterprise Billing provides a mechanism to separate enterprise data usage from personal data usage. This enables enterprises to compensate their employees for work expenditures, particularly in BYOD cases, or to only pay for work-related data in COPE cases.

The Knox platform supports Enterprise Billing on Knox version 2.2 and above, and requires EMM support.

Enterprises configure two *Access Point Name* (APN) gateways. One APN is for data associated with enterprise-approved apps, and a different APN is for all other personal data. Enterprises must first register with a network operator's enterprise billing service. Once a new APN is provisioned for business use, Knox Workspace can be enabled for that dedicated APN. IT admins can also select individual apps inside or outside workspace to use data over the enterprise APN.

Enterprise billing with a dedicated APN can:

- Separate data usage over the mobile Internet for legacy 2G/3G/4G connections
- Route data traffic from the workspace over the enterprise APN
- Provide the capability to select individual apps inside or outside the Knox Workspace to use data over the enterprise APN

The enterprise APN can also be configured to allow or deny roaming. When roaming is enabled, personal data is routed through the default APN, and enterprise data is routed through a dedicated enterprise APN. By default, roaming over the enterprise APN is disabled. When a user is roaming in a single Packet Data Protocol (PDP) network, all enterprise apps are automatically routed to the personal APN for work continuity.

If enterprise apps use a network VPN connection, the VPN profile can be configured to route data through the enterprise APN. Dual SIM devices can also be enabled for Knox Enterprise Billing. The primary, or first SIM slot, is automatically selected to configure an APN and activate Enterprise Billing on the device.

To avoid the personal use of a SIM card, IT admins can lock the SIM card with a unique PIN combination. This ensures the SIM can only be used for enterprise billing on the authorized device. In addition, dedicated enterprise APNs are restricted, and APN settings are not visible or editable on the device.

Users can check personal and enterprise data usage on a Knox device by navigating to the Settings menu. To view data usage, employees navigate to Settings > Data Usage > Mobile Tab (personal) or Enterprise Tab (work).

Endnotes

[1] Kaspersky Lab, “2016 saw 8.5 million mobile malweare attacks," February 28th, 2017. http://www.techrepublic.com/article/report-2016-saw-8-5-million-mobile-malware-attacks-ransomwareand-iot-threats-on-the-rise/

[2] Ibid

[3] Nielsen, “The Digital Consumer,” October 2013, p. 8. http://www.nielsen.com/content/dam/corporate/us/en/reports-downloads/2014%20Reports/the-digital-consumer-report-feb-2014.pdf

[4] Consumer Reports, “Smart phone thefts rose to 3.1 million last year, Consumer Reports finds,” May 2014. http://www.consumerreports.org/cro/news/2014/04/smart-phone-thefts-rose-to-3-1-million-last-year/index.htm

[5] FCC, “Report of Technological Advisory Council (TAC) Subcommittee on Mobile Device Theft Prevention (MDTP),” December 2014, p. 22. http://transition.fcc.gov/bureaus/oet/tac/tacdocs/meeting12414/TAC-MDTP-Report-v1.0-FINAL-TAC-version.pdf

[6] Workshare, “Data Guardian: Detecting Business Risk 2014," p. 14-16. https://d3liiczouvobl1.cloudfront.net/uploads/refinery/resource/file_name/251/Workshare_-_Data_Guardian_-_Detecting_Business_Risk_2014.pdf

About Samsung Electronics Co., Ltd.

Samsung Electronics Co., Ltd. is a global leader in technology, opening new possibilities for people everywhere. Through relentless innovation and discovery, we are transforming the worlds of televisions, smartphones, personal computers, printers, cameras, home appliances, LTE systems, medical devices, semiconductors and LED solutions. We employ 236,000 people across 79 countries with annual sales exceeding KRW 201 trillion. To discover more, please visit www.samsung.com

For more information about Samsung Knox, visit www.samsungknox.com

Samsung Electronics Co., Ltd.
416, Maetan 3-dong, Yeongtong-gu
Suwon-si, Gyeonggi-do 443-772, Korea

Version	Date
Samsung Knox Security Solution V2.2	May 2, 2017
Samsung Knox Security Solution V2.1	December 8, 2016
Samsung Knox Security Solution V2.0	Nov. 17, 2016
Samsung Knox Security Solution V1.12	September 22, 2016

Acronyms

AES	Advanced Encryption Standard
AOSP	Android Open Source Project
CAC	U.S. Common Access Card
CCM	Client Certificate Management
CESG	Communications and Electronic Security Group
CMK	Container Master Key
COBO	Corporate Owned Business Only
COPE	Corporate-Owned Personally Enabled
DAC	Discretionary Access Control
DAR	Data-at-Rest
DISA	U.S. Defense Information Systems Agency
DIT	Data-in-Transit
DRK	Device Root Key
DUHK	Device-Unique Hardware Key
FIPS	Federal Information Processing Standard
IAM	Identity and Access Management
IPC	Inter Process Communication
KEA	Knox Enabled App
MAC	Mandatory Access Control
MAM	Mobile Application Management
MCM	Mobile Container Management
MDM	Mobile Device Management
MMU	Memory Management Unit
NFC	Near Field Communication

Acronyms

NIST	National Institute of Standards and Technology
ODE	On-Device Encryption
PKCS	Public Key Cryptography Standards
PKM	Periodic Kernel Measurement
RKP	Real-time Kernel Protection
RP	Rollback Prevention
SBU	Sensitive But Unclassified
SDP	Sensitive Data Protection
SEAMS	SE for Android Manager Service
SE for Android	Security Enhancements for Android
SE Linux	Security Enhanced Linux
SRG	Security Requirements Guide
SSBK	Samsung Secure Boot Key
SSO	Single Sign-On
STIGs	Security Technical Implementation Guides
TIMA	TrustZone-based Integrity Measurement Architecture
VPN	Virtual Private Network

EXHIBIT 3

Android security maximized by Samsung KNOX

Safeguard enterprise mobility with tightly integrated security, compliance, and control features

SAMSUNG

Contents

Google Android™ Lollipop 3

Samsung KNOX 3

KNOX is always vigilant 3

- Time of design 3
- Time of manufacture 4
- Boot-time defenses 4
- Load-time defenses 5
- Run-time defenses 5
- Update-time defenses 5

Application-level security mechanisms 6

Independently certified 7

Security that fits your existing IT infrastructure 8

Android security maximized by Samsung KNOX 9

Comprehensive enterprise mobile security and productivity 9

Google and Samsung are both committed to mobile enterprise security, each bringing its own considerable expertise on protecting devices and data.

Google Android™ Lollipop

The Lollipop release improves the default security of the Android platform with new security features including:

- Verified Boot defenses against unauthorized modification of the operating system during the boot process.
- Basic VPN functionality for secure connections to enterprise networks.
- Google Safe Browsing technology to block phishing and malware attacks.

Google supports and even encourages device manufacturers, including Samsung, to build upon this solid base to provide solutions that fully address the security issues facing enterprise customers:

- Regulatory compliance
- Liability
- Risk tolerance
- Peace of mind

As the clear leader in security and enterprise readiness among all Android OEMs, Samsung KNOX builds on Lollipop to deliver a comprehensive security solution that addresses these real-world issues for the most demanding enterprise customers.

Samsung KNOX

KNOX is Samsung's defense-grade mobile security platform built into its newest devices. Just turn the device on and you're protected.

Cyber-attacks are generally designed to exploit weaknesses in device software implementation and architecture, or the attack surface. KNOX is designed to minimize the attack surface of devices by:

- Fortifying weaknesses that known attacks have commonly exploited in the past.
- Anticipating and defending against other more insidious categories of attacks.

With large classes of common attacks rendered ineffective against properly configured KNOX-protected devices, would-be attackers are forced to quit or attempt increasingly sophisticated attacks that require more time, money, and expertise.

KNOX is always vigilant

To combat attacks, KNOX establishes defense mechanisms at the time of design, time of manufacture, boot time, software load time (from disk to RAM), and run time as described in the following sections:

Time of design

By design, Samsung KNOX fully leverages the hardware Trusted Execution Environment (TEE) capabilities found in Samsung's flagship mobile devices, as well as many others. Without a TEE or equivalent, secure computing cannot be meaningfully achieved. For example, TEE uses ARM® TrustZone®.

Warranty bit

The KNOX warranty bit is a one-time programmable fuse that is blown when evidence of tampering is detected of bootloaders or the kernel. Thereafter, the device can never run Samsung KNOX, access to the Device Root Key, and access in the TrustZone secure world is revoked. In addition, users cannot access enterprise data on the device.

Time of manufacture

Samsung manufactures and configures its devices in its own factories. This means that Samsung has total control over the state of the device software leaving the factory.

In addition to provisioning the software, Samsung provisions each device with certain cryptographic data upon which nearly all higher-level security processes depend. These include a Device Root Key (unique per unit manufactured) and a Samsung Secure Boot Key (unique to Samsung, but the same on all Samsung devices).

Other device manufacturers that outsource hardware cannot guarantee the same end-to-end control of these critical security elements.

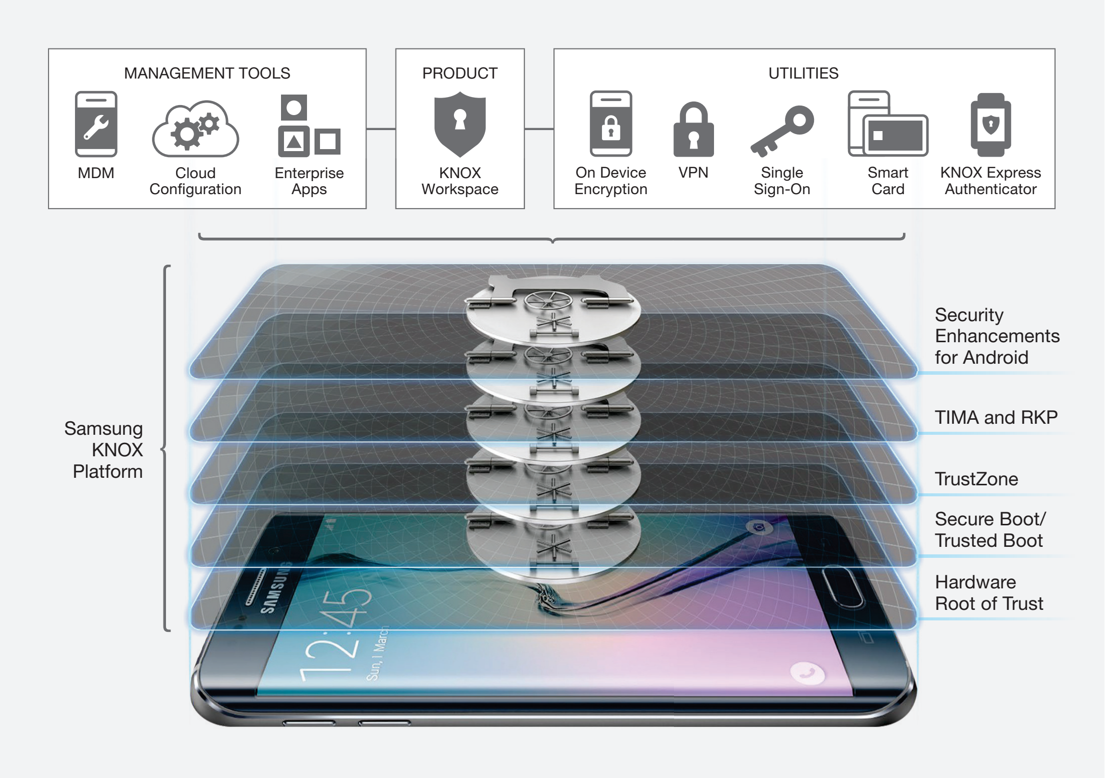


Figure 1. Samsung KNOX makes Android secure for enterprises

Boot-time defenses

One of the most fundamental requirements of mobile security is to ensure the authenticity and integrity of the software that is allowed to run on the device. This includes the stock operating system as well as all the modules that the OEM is required to provide.

KNOX employs its Secure Boot and Trusted Boot to ensure that they verify both the authenticity and integrity of the bootloader modules and the Android kernel. It does this by sequentially verifying chunks of code against previously-generated cryptographic signatures stored in secure memory of the TEE.

Load-time defenses

Smartphones and tablets have a large amount of preloaded system software beyond the operating system kernel. The size of this system software makes it impractical to verify its integrity and authenticity at boot time as it would introduce unacceptable start-up delay for the user.

Like all Lollipop devices, KNOX employs a technique called DM-Verity to ensure the integrity of system software not covered by the boot time checking described earlier. However, Samsung's implementation of DM-Verity differs from stock Lollipop in several important ways:

1. Modified to accommodate the real-world need for devices to accept firmware over-the-air (FOTA) software updates.
2. File-based instead of block-based to support carrier-specific and region-specific software builds
3. Its use is optional for non-enterprise consumer users of devices.

With Secure Boot, Trusted Boot, and DM-Verity, enterprises can feel confident that the software is authentic and uncompromised by the time it is loaded into RAM for execution.

Run-time defenses

Some more sophisticated attacks seek to compromise the system or intercept data at run time.

Periodic Kernel Measurement (PKM)

TrustZone-based Integrity Measurement Architecture (TIMA) PKM performs continuous periodic monitoring of the kernel to detect if legitimate kernel code and data have been modified unexpectedly.

Real-time Kernel Protection (RKP)

TIMA RKP performs ongoing real-time monitoring of the operating system from within TrustZone to prevent tampering of the kernel. RKP intercepts critical kernel events that are then inspected in TrustZone. If an event is determined to have unauthorized impact on the integrity of the OS kernel, RKP either stops the event, or logs an attestation record that tampering is suspected. This protects against malicious modifications and injections to kernel code, including those that coerce the kernel into corrupting its own data.

Attestation

Attestation reads the Trusted Boot collected measurement data and the fuse value, then combines the data in a proprietary way to produce an Attestation verdict that can be requested on-demand by the enterprise's MDM, typically before creating the KNOX Workspace. This verdict, essentially a coarse indication that tampering is suspected, is returned to the requesting MDM. The attestation verdict is cryptographically signed to ensure its integrity and authenticity.

Update-time defenses

Rollback Prevention blocks the device from loading an approved but old version of boot components during Trusted Boot. Old versions of software may contain known vulnerabilities that attackers can exploit. Rollback prevention checks the version of the bootloader and kernel during both Trusted Boot and updates, and blocks these processes from continuing if versions are unacceptably old. The lowest acceptable version of the bootloader is stored in secure hardware fuses at the time of manufacture, and the lowest acceptable version of the kernel is stored in the bootloader itself.

Table 1 - Summary of KNOX Defenses Mechanisms

Feature	Description
Hardware	
Samsung Secure Boot Key	Verifies that all firmware is from Samsung before allowing the device to boot.
Device Root Key	Provides a unique key per device that is used to perform cryptographic operations (authentication and encryption) associated with that specific device.
Warranty Bit	Creates a one-time, writeable hardware “fuse” used to flag devices whose system software has been replaced, in part or in full, either intentionally or maliciously.
Rollback Prevention Fuses	Set at manufacturing time in the Samsung factory to prevent old firmware versions from overwriting newer ones.
Bootloader	
Secure Boot	Ensures the integrity of each component of the boot software until just before the Android kernel is launched. (Uses the Samsung Secure Boot Key). If anything else tries to run outside of the valid, trusted sequence, the boot process terminates.
Trusted Boot	Builds upon Secure Boot to ensure the end-to-end integrity and consistency of boot software—including the kernel—for the entire boot process. Any evidence of tampering is permanently logged.
Rollback Prevention	Uses rollback prevention fuses to ensure that an old (but valid) firmware image cannot overwrite more recent images.
TrustZone	
Periodic Kernel Measurement	Performs continuous periodic monitoring of the kernel to detect if kernel code or data has been modified by malicious software.
Real-time Kernel Protection	Maintains runtime integrity by monitoring critical events that occur in the Android kernel and enforces protection of the kernel code so that it cannot be moved, changed, or amended.
Attestation	Allows a device to attest to a remote server, such an MDM server, that it has loaded authorized images during boot time.
TIMA KeyStore	Provides applications with services for generating and maintaining cryptographic keys. The keys are further encrypted with the device-unique hardware key that can only be decrypted by the hardware inside TrustZone.
Client Certificate Management	Enables storage and retrieval of digital certificates for encryption, decryption, signing, verification, and other operations.
Fingerprint Authentication	Requires apps to use fingerprints as a primary or two-factor authentication with checks performed in the TrustZone.

Application-level security mechanisms

Once KNOX verifies device integrity, the next hurdle is to address the security of the applications and data. Organizations must ensure that data stored on devices cannot be breached or shared inappropriately, and that applications accessing company information can be used only for corporate purposes.

Android Lollipop provides baseline data and application security. There is an Android KeyStore, which can encrypt and store cryptographic keys. Android Work Profiles can isolate personal apps and data from enterprise apps and data. You also get basic VPN functionality for a secure connection to corporate resources, as well as a way to identify harmful apps by using the Google Safe Browsing scanner.

Samsung KNOX significantly builds and improves upon this foundation to provide enterprise-class mobile application and data security. Cryptographic keys and other important security data is stored hardware in TEE – allowing only authorized users to access confidential data. Encryption and container technologies also keep data and applications safe – preventing corporate data from being shared inappropriately.

Table 2 - Application-level security features

Feature	Description
TIMA KeyStore	Improves upon the standard Android KeyStore by denying access to its contents when Trusted Boot or Warranty Bit reports that the device has potentially been compromised. Stored keys cannot be cloned for use on other devices.
TIMA Client Certificate Management (CCM)	Enables cryptographic keys to be sequestered in a secure area of the device, so that private key information is never exposed to the Android operating system.
Workspace	Offers a defense-grade, dual-persona container product designed to separate, isolate, encrypt, and protect work data from attackers. This work/play environment ensures work data and personal data are separated and that only the work container can be managed by the enterprise. Personal information such as photos and messages are not managed or controlled by the IT department. Once activated, the KNOX Workspace product is tightly integrated into the KNOX platform.
Sensitive Data Protection	Builds upon Workspace encryption, defining a sensitive class of data, which the device cannot decrypt without user intervention. TIMA KeyStore used to manage cryptographic keys.
VPN Framework	Adds FIPS 140-2 certified cryptographic algorithms, or the option to use CCM to manage cryptographic keys, to establish secure VPN connections to corporate network resources. Integrates with Workspace to assure that applications route network traffic through approve channels.
SSO Framework	Enhances authentication of Workspace apps by providing a common framework. Backed by TIMA KeyStore and CCM.
On-Disk Encryption	Uses a derivative of the Device Root Key to strengthen Android's On-Disk Encryption feature, ensuring that copying the raw data from one device to another is not possible.
Hardware Attestation	Uses a derivative of the Device Root Key, plus measurements collected from Trusted Boot, Warranty Bit, and RKP, to securely attest the state of the device to a remote server.
No Mandated Cloud Connection	Eliminates the requirement to connect an employee's device to a third-party cloud server (Google Cloud). KNOX license server can be deployed as an on-premises instance to avoid any cloud connection.

Independently certified

Because of these security capabilities that allow users to trust their data, Samsung KNOX has been awarded multiple, internationally recognized security certifications from governments around the world.

Table 3 - Independent Security Certifications

Country	Certification	Issued by
USA/Canada	Federal Information Processing Standard 140-2 Certification – Level 1 certification for both data-at-rest (DAR) and data-in-transit.	National Institute of Standards and Technology (NIST)
USA	Security Technical Implementation Guides (STIGs), DISA Approved Product List	Defense Information Systems Agency (DISA)
USA	Common Criteria Certification for Mobile Device Fundamental Protection Profile (MDFPP)	National Information Assurance Partnership (NIAP)
USA	US Department of Defense Approved Products List	National Information Assurance Partnership (NIAP)
UK	End User Devices (EUD) Security Guidance	Communications and Electronics Security Group (CESG)
Finland	Finnish National Security Auditing Criteria (KATAKRI II)	Finnish Communications Regulatory Authority (FICORA)
Australia	Protection Profile for Mobile Device Fundamentals	Australian Signals Directorate (ASD)

Security that fits your existing IT infrastructure

Enterprises large and small have diverse needs when it comes to device management. Samsung KNOX provides enterprises comprehensive control to configure the Workspace to their needs using an extensive set of more than 1500 Mobile Device Management (MDM) APIs.

Samsung KNOX also provides utilities that allow ready deployment in enterprises such as per-application VPN controls, a smartcard framework, and Single Sign-on (SSO) integration with Microsoft Active Directory. These features enable Samsung KNOX to easily integrate into any enterprise.

Table 4 - Defense-grade security features

Enterprise Mobility Infrastructure	
Identity/Email Registration	KNOX allows you to avoid registering an email address with Google to manage user identity on a device.
Exchange/ActiveSync	KNOX supports use of Exchange/ActiveSync for messaging.
LDAP Support	KNOX has explicit built-in support for LDAP account configuration and credentials. KNOX also supports Microsoft Active Directory.
VPN	KNOX provides tailored support for a growing list of industry-leading VPN clients from Cisco, Juniper, Mocana, F5, OpenVPN, StrongSwan and more. The VPN framework also allows easy adoption of additional VPN solutions.
Third-party Container Support	KNOX enables third-party container solutions to benefit from various KNOX security features.
Firewall Configuration	KNOX provides APIs to configure firewall policies.
User Experience	
Multiple Simultaneous Containers	KNOX supports multiple simultaneous containers/profiles, while Android alone can only accommodate one profile.
Kiosk and Container-Only Mode	KNOX kiosk and container-only mode allow clear work/personal boundaries.
Container UX	KNOX allows the user or enterprise to choose between three different user experiences, depending on the needs of the individual or organization: Classic (separate, isolated UX), Folder (pop-up UX), or Full Screen (continuous feed).
Mobile Device Management	
Onboarding/Enrollment	KNOX never requires devices to connect to Samsung servers to authenticate or register an identity for onboarding or enrollment.
Device Control	KNOX includes a fully integrated set of management tools that offer deep device control of security, usability, hardware, and application policies. KNOX also offers easy integration with third-party Mobile Device Management solutions.
My KNOX	Individual mobile professionals can use Samsung My KNOX to separate work and personal data separate.
Application Management	KNOX supports Google Play, Samsung App Store, KNOX marketplace, MDM solutions, and manual side-loading to deploy applications. All transactions are 100% anonymous in an enterprise-managed model. Android requires you use Google Play for every app management transaction and prohibits side-loading.
Telephony	KNOX enables policies to block incoming/outgoing voice and SMS.
Password Policy	KNOX extends Android password policies with more granular control over precise requirements for character sets, repeated characters, refresh periods, and reuse.
User Privacy	KNOX Workspace limits the employer's visibility into and control of the Workspace, putting the non-Workspace data and apps beyond the reach of the employer.

Comprehensive enterprise mobile security and productivity

When implementing your enterprise mobile strategy, Android devices alone are not enough. While the security features in Lollipop have improved Android's position with competitors such as iPhone, most enterprises find that their security and compliance requirements are not met.

Samsung KNOX augments Lollipop security features to produce a tightly integrated and holistic security architecture. By enabling all of the security, compliance, and control features enterprises require, organizations can use Samsung KNOX to enable worker productivity while also protecting corporate assets.

About Samsung Electronics Co., Ltd.

Samsung Electronics Co., Ltd. is a global leader in technology, opening new possibilities for people everywhere. Through relentless innovation and discovery, we are transforming the worlds of TVs, smartphones, tablets, PCs, cameras, home appliances, printers, LTE systems, medical devices, semiconductors and LED solutions. We employ 286,000 people across 80 countries with annual sales of US $216.7 billion. To discover more, please visit www.samsung.com.

For more information

For more information about Samsung Enterprise Mobility and Samsung KNOX, visit: www.samsung.com/enterprise and www.samsung.com/knox

SAMSUNG

Samsung Knox

EXHIBIT 4

Beyond basic Android

Security with Samsung KNOX

SAMSUNG
BUSINESS

Contents

Google Android™ Lollipop and Android for Work: A starting foundation for mobile security 3

Enterprise-ready security built-in with Samsung KNOX 3

KNOX is always on, always vigilant 3

Secure by design 3

Controlling the manufacturing process 4

Application-level security mechanisms 7

Independent security certifications 8

Advanced security that fits your existing IT infrastructure 8

Comprehensive enterprise mobile security and productivity 9

Appendix: Feature comparison of Samsung KNOX and Android for Work 10

Google and Samsung are committed to mobile enterprise security, each bringing its own considerable expertise to the challenge of protecting devices and data from ever-growing threats and attacks. Backed by years of research, Samsung KNOX offers unsurpassed levels of mobile security that make Samsung devices truly enterprise-ready right out of the box.

Google Android™ Lollipop and Android for Work: A starting foundation for mobile security

Coinciding with the release of Android Lollipop, Android for Work improves the basic security of the Android platform with new security features including:

- Verified Boot of the operating system during the boot process.
- Limited VPN functionality for secure connections to enterprise networks.
- Google Safe Browsing technology.

Google relies on device manufacturers such as Samsung to expand basic security and provide integrated security solutions for enterprises addressing issues, including:

- Regulatory compliance
- Liability
- Risk tolerance

As the leader in security and enterprise readiness, Samsung KNOX delivers the most comprehensive solution of all Android device manufacturers. KNOX mobile security provides unmatched protection against mobile threats from the moment you turn on the device.

Enterprise-ready security built in with Samsung KNOX

KNOX is the defense-grade mobile security platform built into all of Samsung's newest devices.

Cyber attacks exploit weaknesses in device software and architecture. KNOX minimizes the attack surface with the following:

- Security checks starting from the hardware level at power up.
- TrustZone-based Integrity Measurement Architecture (TIMA) Real-time Kernel Protection.
- TIMA Periodic Kernel Measurement.
- Reporting (Attestation) of tampering with the OS, apps, or device architecture.

KNOX is always on, always vigilant

KNOX establishes its first line of defense with Secure Boot and Trusted Boot architecture, coupled with chip-level security. These hardware, firmware, and software checks are followed by multiple checks to ensure that each part of the operating system hasn't been tampered with before corporate applications can run. If any of those checks fail, KNOX reports the threat to the Mobile Device Management (MDM) system.

Secure by design

KNOX fully leverages the hardware Trusted Execution Environment (TEE) of ARM® TrustZone® capabilities found in Samsung's flagship mobile devices. KNOX provides strong guarantees for the protection of enterprise data by building a hardware-rooted trusted environment. A trusted environment ensures that enterprise-critical operations, such as decryption of enterprise data, can occur only when the device is proven to be in an allowed state. For many pieces of device software, such as the kernel and TrustZone apps, the allowed state is represented by the cryptographic signature of each piece of software. A trusted environment is hardware-rooted if both the cryptographic keys and code used to compute these signatures are tied back to unmodifiable values stored in hardware.

Controlling the manufacturing process

Samsung manufactures and configures all its devices in its own factories, providing Samsung with complete control over the devices and software before they leave the factory. In addition to provisioning the software on the devices, Samsung provisions each device with the cryptographic keys, such as the Device-Unique Hardware Key (DUHK) and the Samsung Secure Boot Key (SSBK). The Secure Boot process uses the SSBK to verify whether each boot component it loads is approved. Data encrypted by the DUHK becomes bound to the device, because it cannot be decrypted on any other device.

Other device manufacturers that outsource hardware cannot guarantee the same end-to-end control of these critical security elements. The additional steps Samsung takes to protect the manufacturing process surpass what other device manufacturers and OEMs can provide to their customers.

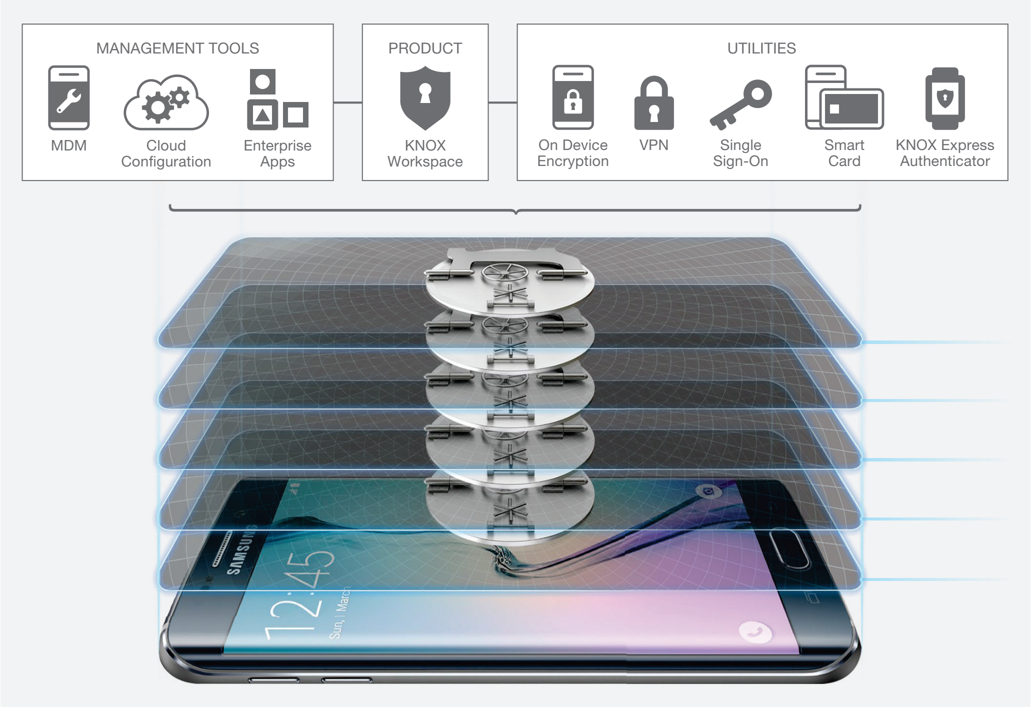


Figure 1. Samsung KNOX makes Android secure for enterprises

Warranty Fuse

The KNOX warranty bit is a one-time programmable fuse that indicates whether the device has ever been booted into an unapproved state. If the Trusted Boot process detects that non-approved components are used, or if certain critical security features such as SELinux are disabled, it sets the fuse. Once the fuse is blown, the device can never run Samsung KNOX; access to the DUHK and DRK in the TrustZone Secure World is revoked, and the enterprise's data on the device cannot be recovered.

Boot-time defenses

One of the most fundamental requirements of mobile security is to ensure the authenticity and integrity of the software allowed to run on the device. This includes the operating system as well as all the modules that the OEM is required to provide on the device. KNOX employs Secure Boot and its own Trusted Boot layers to verify the authenticity and integrity of bootloader modules and the Android kernel during boot. It does this by verifying chunks of code against previously-generated cryptographic signatures stored in secure memory of the TEE.

Load-time defenses

Today's smartphones and tablets have a large amount of preloaded system software beyond the operating system kernel. The size of this system software makes it impractical to verify its integrity and authenticity at boot time, because it would introduce unacceptable start-up delay for the user.

KNOX uses an enhanced version of the stock implementation of DM-Verity included with Android Lollipop to ensure the integrity of system software not covered by the boot time checks. Samsung's implementation of DM-Verity differs from standard Lollipop in several important ways:

1. Modified to accommodate the real-world need for devices to accept firmware over-the-air (FOTA) software updates.
2. File-based checking instead of block-based to support carrier-specific and region-specific software builds.
3. Optional for non-enterprise consumer users of devices.

With Secure Boot, Trusted Boot, and DM-Verity, enterprises can be confident that the all device software is authentic and uncompromised.

Run-time defenses

More sophisticated attacks can compromise the system or intercept data at run time. These are the most difficult attacks to prevent, but KNOX gives users and enterprises confidence that malicious apps or code can't run on KNOX-protected devices. The combination of Periodic Kernel Measurement, Real-time Kernel Protection, and device Attestation protects devices beyond the basic levels provided by Android for Work through other OEMs. KNOX gives enterprise IT Admins the tools to track down and prevent attacks while they can be easily contained.

Periodic Kernel Measurement (PKM)

TIMA PKM performs continuous periodic monitoring of the kernel to detect if legitimate kernel code and data have been modified by malicious software. In addition, TIMA also monitors key Security Enhancements (SE) for Android data structures in operating-system kernel memory to detect malicious attacks that corrupt them and potentially disable SE for Android.

Real-time Kernel Protection (RKP)

TIMA RKP performs ongoing, real-time monitoring of the operating system from within TrustZone to prevent kernel tampering. RKP protects against malicious modifications and injections to kernel code, including those that coerce the kernel into corrupting its own data. RKP intercepts and inspects critical kernel events in the TrustZone, and if an event is determined to have unauthorized impact on the integrity of the OS kernel, RKP either stops the event or logs an attestation record that tampering is suspected.

Attestation

Attestation reads the data collected by Trusted Boot and fuse values and combines them to produce an Attestation verdict that can be requested on-demand by the enterprise's Mobile Device Management (MDM) system (typically before creating the KNOX Workspace). This verdict, a coarse indication that tampering is suspected, is returned to the requesting MDM. The Attestation verdict is cryptographically signed to ensure its integrity and authenticity.

RKP and PKM are essential to protect against future threats to Android and devices that haven't yet been uncovered or predicted. Flagging activities that we know are suspicious is the first step in identifying and preventing security breaches. These real-time checks, together with Attestation, give enterprises confidence that devices are continuously monitored for breaches, and that IT is alerted if corporate devices have been compromised.

Update-time defenses

Rollback Prevention blocks the device from loading an approved but old version of boot components during Trusted Boot. Old versions of software may contain known vulnerabilities that attackers can exploit. Rollback prevention checks the version of the bootloader and kernel during both Trusted Boot and updates, and blocks these processes from continuing if versions are unacceptably old. The lowest acceptable version of the bootloader is stored in secure hardware fuses at the time of manufacture, and the lowest acceptable version of the kernel is stored in the bootloader itself.

Table 1 - Summary of KNOX Defense Mechanisms

Feature	Description
Hardware	
Samsung Secure Boot Key	Verifies that all firmware is from Samsung before allowing the device to boot.
Device Root Key	Provides a unique key per device that is used to perform cryptographic operations (authentication and encryption) associated with that specific device.
Warranty Bit	Creates a one-time, writeable hardware "fuse" used to flag devices whose system software has been replaced, in part or in full, either intentionally or maliciously.
Rollback Prevention Fuses	Set at manufacturing time in the Samsung factory to prevent old firmware versions from overwriting newer ones.
Bootloader	
Secure Boot	Ensures the integrity of each component of the boot software until just before the Android kernel is launched. (Uses the Samsung Secure Boot Key). If anything else tries to run outside of the valid, trusted sequence, the boot process terminates.
Trusted Boot	Builds upon Secure Boot to ensure the end-to-end integrity and consistency of boot software—including the kernel—for the entire boot process. Any evidence of tampering is permanently logged.
Rollback Prevention	Uses rollback prevention fuses to ensure that an old (but valid) firmware image cannot overwrite more recent images.
TrustZone	
Periodic Kernel Measurement	Performs continuous periodic monitoring of the kernel to detect if kernel code or data has been modified by malicious software.
Real-time Kernel Protection	Maintains runtime integrity by monitoring critical events that occur in the Android kernel and enforces protection of the kernel code so that it cannot be moved, changed, or amended.
Attestation	Allows a device to attest to a remote server, such an MDM server, that it has loaded authorized images during boot time.
TIMA KeyStore	Provides applications with services for generating and maintaining cryptographic keys. The keys are further encrypted with the device-unique hardware key that can only be decrypted by the hardware inside TrustZone.
Client Certificate Management	Enables storage and retrieval of digital certificates for encryption, decryption, signing, verification, and other operations.
Fingerprint Authentication	Requires apps to use fingerprints as a primary or two-factor authentication with checks performed in the TrustZone.

Application-level security mechanisms

Once KNOX verifies device integrity, it begins testing application and data security for threats. Organizations must trust that data stored on devices cannot be breached or shared inappropriately, and that applications accessing company information can be used only for corporate purposes. KNOX also protects against data attacks with sophisticated, automatic VPN capabilities.

TIMA KeyStore

Samsung KNOX builds and improves upon the basic Android KeyStore foundation to provide enterprise class mobile application and data security with a hardware-based TIMA KeyStore. KNOX uses a TrustZone-based TIMA KeyStore for cryptographic key storage. Keys stored in the TIMA KeyStore can be accessed only if the measurements collected by Trusted Boot match the expected golden measurement values, and the warranty fuse has not been set. If the device fails these critical integrity checks, then applications cannot access enterprise data. The TIMA KeyStore protects sensitive data, including keys, by hardware-derived keys in TrustZone. Hardware-bound security ensures protected data can only be decrypted on the same device by TIMA KeyStore.

Automatic and per-app VPN settings

While Android for Work allows for some granular control for app-by-app VPN settings, KNOX extends this capability to a broader range of supported VPNS, using different VPNs for different apps, and even separating out the data used by corporate applications for easier accounting for BYOD and Corporately Owned, Personally Enabled (COPE) devices. Recognizing the increasing complexity of enterprise IT, Samsung has built sophisticated VPN support into KNOX to make integrating KNOX into IT environments easier to manage.

Table 2 - Application-level security features

Feature	Description
TIMA KeyStore	Improves upon the standard Android KeyStore by denying access to its contents when Trusted Boot or Warranty Bit reports that the device has potentially been compromised. Stored keys cannot be cloned for use on other devices.
TIMA Client Certificate Management (CCM)	Enables cryptographic keys to be sequestered in a secure area of the device, so that private key information is never exposed to the Android operating system.
Workspace	Offers a defense-grade, dual-persona container product designed to separate, isolate, encrypt, and protect work data from attackers. This work/play environment ensures work data and personal data are separated and that only the work container can be managed by the enterprise. Personal information such as photos and messages are not managed or controlled by the IT department. Once activated, the KNOX Workspace product is tightly integrated into the KNOX platform.
Sensitive Data Protection	Builds upon Workspace encryption, defining a sensitive class of data, which the device cannot decrypt without user intervention. TIMA KeyStore is used to manage cryptographic keys.
VPN Framework	Adds FIPS 140-2 certified cryptographic algorithms, or the option to use CCM to manage cryptographic keys, to establish secure VPN connections to corporate network resources. Integrates with Workspace to assure that applications route network traffic through approve channels.
SSO Framework	Enhances authentication of Workspace apps by providing a common framework. Backed by TIMA KeyStore and CCM.
On-Disk Encryption	Uses a derivative of the Device Root Key to strengthen Android's On-Disk Encryption feature, ensuring that copying raw data from one device to another is not possible.
Hardware Attestation	Uses a derivative of the Device Root Key, plus measurements collected from Trusted Boot, Warranty Bit, and RKP, to securely attest the state of the device to a remote server.
No Mandated Cloud Connection	Eliminates the requirement to connect an employee's device to a third-party cloud server (Google Cloud). KNOX license server can be deployed as an on-premises instance to avoid any cloud connection.

Independent security certifications

Samsung KNOX has been awarded multiple, internationally recognized security certifications from government authorities around the world. These certifications help set Samsung apart from other OEMs and help validate the security claims that Samsung makes for its devices.

Table 3 - Independent Security Certifications

Country	Certification	Issued by
USA/Canada	Federal Information Processing Standard 140-2 Certification – Level 1 certification for both data-at-rest (DAR) and data-in-transit.	National Institute of Standards and Technology (NIST)
USA	Security Technical Implementation Guides (STIGs), DISA Approved Product List	Defense Information Systems Agency (DISA)
USA	Common Criteria Certification for Mobile Device Fundamental Protection Profile (MDFPP)	National Information Assurance Partnership (NIAP)
USA	US Department of Defense Approved Products List	National Information Assurance Partnership (NIAP)
UK	End User Devices (EUD) Security Guidance	Communications and Electronics Security Group (CESG)
Finland	Finnish National Security Auditing Criteria (KATAKRI II)	Finnish Communications Regulatory Authority (FICORA)
Australia	Protection Profile for Mobile Device Fundamentals	Australian Signals Directorate (ASD)

Advanced security that fits your existing IT infrastructure

Enterprises large and small have diverse device management needs. KNOX provides enterprises with a comprehensive set of tools for configuring the secure KNOX Workspace to their needs, including more than 1500 MDM APIs.

KNOX provides tools for rapid enterprise deployment, such as per-application VPN controls, a smartcard framework, Single Sign-on (SSO) integration with Microsoft Active Directory, KNOX Mobile Enrollment (for bulk loading devices into an MDM) and Enterprise Billing for tracking employee data usage on BYOD and COPE devices.

Unlike Android for Work, KNOX allows access to a broader range of approved app stores in addition to Google Play. If you have your own in-house apps, KNOX allows enterprise apps to be securely side-loaded onto devices. Android for Work requires private, proprietary apps to be uploaded to private Google Play accounts. KNOX allows secure signing and loading of apps into your Workspace without accidentally risking exposure of private apps to the public. Additional features only available with Samsung KNOX are in the table on the following page.

Table 4 - Defense-grade security features

Enterprise Mobility Infrastructure	
Identity/Email Registration	KNOX allows you to avoid registering an email address with Google to manage user identity on a device.
Exchange/ActiveSync	KNOX supports use of Exchange/ActiveSync for messaging.
LDAP Support	KNOX has explicit built-in support for LDAP account configuration and credentials. KNOX also supports Microsoft Active Directory.
VPN	KNOX provides tailored support for a growing list of industry-leading VPN clients from Cisco, Juniper, Mocana, F5, OpenVPN, StrongSwan and more. The VPN framework also allows easy adoption of additional VPN solutions.
Third-party Container Support	KNOX enables third-party container solutions to benefit from various KNOX security features.
Firewall Configuration	KNOX provides APIs to configure firewall policies.
User Experience	
Multiple Simultaneous Containers	KNOX supports multiple simultaneous containers/profiles, while Android alone can only accommodate one profile.
Kiosk and Container-Only Mode	KNOX kiosk and container-only mode allow clear work/personal boundaries.
Container UX	KNOX allows the user or enterprise to choose between three different user experiences, depending on the needs of the individual or organization: Classic (separate, isolated UX), Folder (pop-up UX), or Full Screen (continuous feed).
Mobile Device Management	
Onboarding/Enrollment	KNOX never requires devices to connect to Samsung servers to authenticate or register an identity for onboarding or enrollment.
Device Control	KNOX includes a fully integrated set of management tools that offers deep device control of security, usability, hardware, and application policies. KNOX also offers easy integration with third-party Mobile Device Management solutions.
My KNOX	Individual mobile professionals can use Samsung My KNOX to keep work and personal data separate.
Application Management	KNOX supports Google Play, Samsung App Store, KNOX marketplace, MDM solutions, and manual side-loading to deploy applications. All transactions are 100% anonymous in an enterprise-managed model. Android requires that you use Google Play for every app management transaction and prohibits side-loading.
Telephony	KNOX enables policies to block incoming/outgoing voice and SMS.
Password Policy	KNOX extends Android password policies with more granular control over precise requirements for character sets, repeated characters, refresh periods, and re-use.
User Privacy	KNOX Workspace limits the employer's visibility into and control of the Workspace, putting the non-Workspace data and apps beyond the reach of the employer.

Samsung has worked with MDM, VPN, and application vendors for nearly three years to ensure that KNOX can be deployed in the widest possible number of circumstances.

Comprehensive enterprise mobile security and productivity

When implementing your enterprise mobile strategy, centralizing on Android devices alone is not enough. While the security features in Android for Work have improved Android's position with competitors on other platforms, most enterprises find that their security and compliance requirements are not met.

Samsung KNOX augments Android for Work's security features to produce a tightly integrated and holistic security architecture. By enabling all of the security, compliance, and control features enterprises require, organizations can use Samsung KNOX to enable employee productivity while also protecting corporate assets.

Appendix: Feature comparison of Samsung KNOX and Android for Work

Capability	Samsung KNOX	Android for Work
Silent Install	Using the Samsung KNOX Workspace Mobile Device Management (MDM) APIs, IT admins can install and enable applications automatically. The simplified enrollment process supports the fully automated creation of an enterprise-grade Workspace and provisioning of apps and policies. **KNOX adds:** Samsung KNOX Mobile Enrollment allows IT Admins to stage and enroll hundreds or thousands of employees automatically by configuring device information in the cloud. Samsung also provides a web tool and an application to scan smartphone package bar codes (the device IMEI).	Using the EMM console, IT admins can install, remove, and update apps inside Android for Work.
Application Configuration	KNOX provides the following capabilities to IT admins: • Install and uninstall applications. • Restrict installation and uninstallation of applications. • Disable and enable applications. • Query the current state of an application. • Control application behavior. • Control notifications of applications. • Configure the email client. • Configure the SSL VPN Client for Cisco, F5, and Juniper.	Using the EMM console, IT admins can configure the settings for a particular application. When Android for Work is configured, app settings are pushed to the device.
Secure App Installation from Google Play	With more than 1500 MDM APIs, KNOX gives IT admins control over which apps can be run inside the Workspace, thus eliminating the problem of sideloading of untrusted apps. Additionally, administrators can deploy any app from the Google Play store to the Workspace, or allow users to install the Google Play app inside the Workspace. IT admin can also install applications from a private app store.	Google has introduced a new set of Google Play APIs for EMM providers to enable app management and distribution and control app deployment in Android for Work.
Separate Container for Work Apps	The KNOX Workspace provides an isolated environment and UI for enterprise use, consisting of a separate home screen, launcher, enterprise apps, and widgets. Data owned by apps in the KNOX Workspace is protected by extensive Data At Rest (DAR) protections. IT admins can use KNOX's extensive set of Workspace configuration APIs to provision and configure the Workspace and its DAR protections.	Android for Work provides a secure profile, or container, to Android devices running Android 4.0 and higher.
Data Loss Prevention	KNOX MDM policies can regulate sharing of information between the Workspace and personal apps. This includes sharing of calendar, contacts and notifications. Copy/paste clipboard data is blocked from the Workspace environment to the personal environment, and vice versa. **KNOX adds:** **Sensitive Data Protection.** Any sensitive data received when the Workspace is locked will still be protected by Sensitive Data Protection (SDP). This works by using a public key algorithm in which the private part of the key is maintained in an encrypted partition, and the public part is used to encrypt the new sensitive data. Once the Workspace is unlocked, the data is decrypted with the private key, and re-encrypted using the usual symmetric key, which is guarded by the Container Master Key (CMK). Currently, email subjects, bodies and attachments are marked sensitive. Additionally, the SDP Chamber provides a directory in which all files are automatically marked as sensitive and are protected by SDP.	EMM governance policies manage a user's ability to share into and outside of Android for Work. This includes the ability to block copy/paste or block screen capture for apps inside the managed profile. (Note that copy/paste can be disallowed from the managed profile to the personal profile, but not vice versa.)

Capability	Samsung KNOX	Android for Work
Container VPN	KNOX enables additional modes of granular VPN capabilities both for the Workspace and individual apps. The MDM-configurable KNOX VPN supports multiple concurrent VPN connections allowing for IPSec or SSL VPNs with configurable auto-reconnect and VPN tunnel chaining. The KNOX VPN subsystem also supports other forms of packet processing, including split billing and network access control. **KNOX adds:** Enterprise Billing provides enterprises a mechanism to separate enterprise data usage from personal data usage. This enables enterprises to compensate employees for costs related to work, particularly in BYOD cases, or to pay only work-related data in COPE cases. VPN features of KNOX include: • Administrator-configured System VPN. • Administrator-configured Per-App VPN. • Administrator-configured Workspace VPN. • Multiple concurrent VPN connections. • IPsec and SSL VPN support. • Administrator-configured FIPS and non-FIPS VPN mode. • Common Access Card (CAC)-based authentication. • Always on VPN connections with auto-reconnect. • VPN tunnel chaining.	Android for Work enables VPN capabilities within the managed profile.
Selective Wipe	IT admins can wipe internal and external SD cards and application data. The entire container can be locked when compromised and can be deleted with all its data.	Android for Work enables IT administrators to retire lost or stolen devices and remotely wipe all work data while leaving personal content intact on the device.
Protection Against Malicious App Downloads	The KNOX Workspace isolates enterprise apps and data from personal user apps. Untrustworthy personal user apps outside the Workspace cannot affect the Workspace. **KNOX adds:** Real-time Kernel Protection (RKP) provides three important security benefits: • Prevents running unauthorized privileged code (i.e., code that has the kernel privilege) on the system. • Prevents kernel data from being directly accessed by user processes. This includes preventing double mapping of physical memory that contains critical kernel data into userspace virtual memory. This is an important step to prevent kernel exploits that map kernel data regions into malicious processes where they could be modified by an attacker. • Monitors some critical kernel data structures to verify that they are not exploited by attacks. KNOX Warranty Fuse: The KNOX warranty bit is a one-time programmable fuse that signifies whether the device has ever been booted into an unapproved state. The warranty bit prevents a compromised device from running Samsung KNOX, accessing the Device-Unique Hardware Key (DUHK) and Device Root Key (DRK) in the TrustZone, and accessing any enterprise data. TIMA Attestation allows a device to attest facts about its state to a remote server, such as an MDM server. The attestation message contains measurements about the state of the device. This is evaluated by the MDM to prove the device is in a trusted state, or if there is evidence of tampering.	Android for Work protects business apps and data from issues arising from the user's personal activity outside the profile, such as sideloading web apps.

SAMSUNG Knox

About Samsung Electronics Co., Ltd.

Samsung Electronics Co., Ltd. is a global leader in technology, opening new possibilities for people everywhere. Through relentless innovation and discovery, we are transforming the worlds of TVs, smartphones, tablets, PCs, cameras, home appliances, printers, LTE systems, medical devices, semiconductors, and LED solutions. We employ 286,000 people across 80 countries with annual sales of US $216.7 billion. To discover more, please visit www.samsung.com.

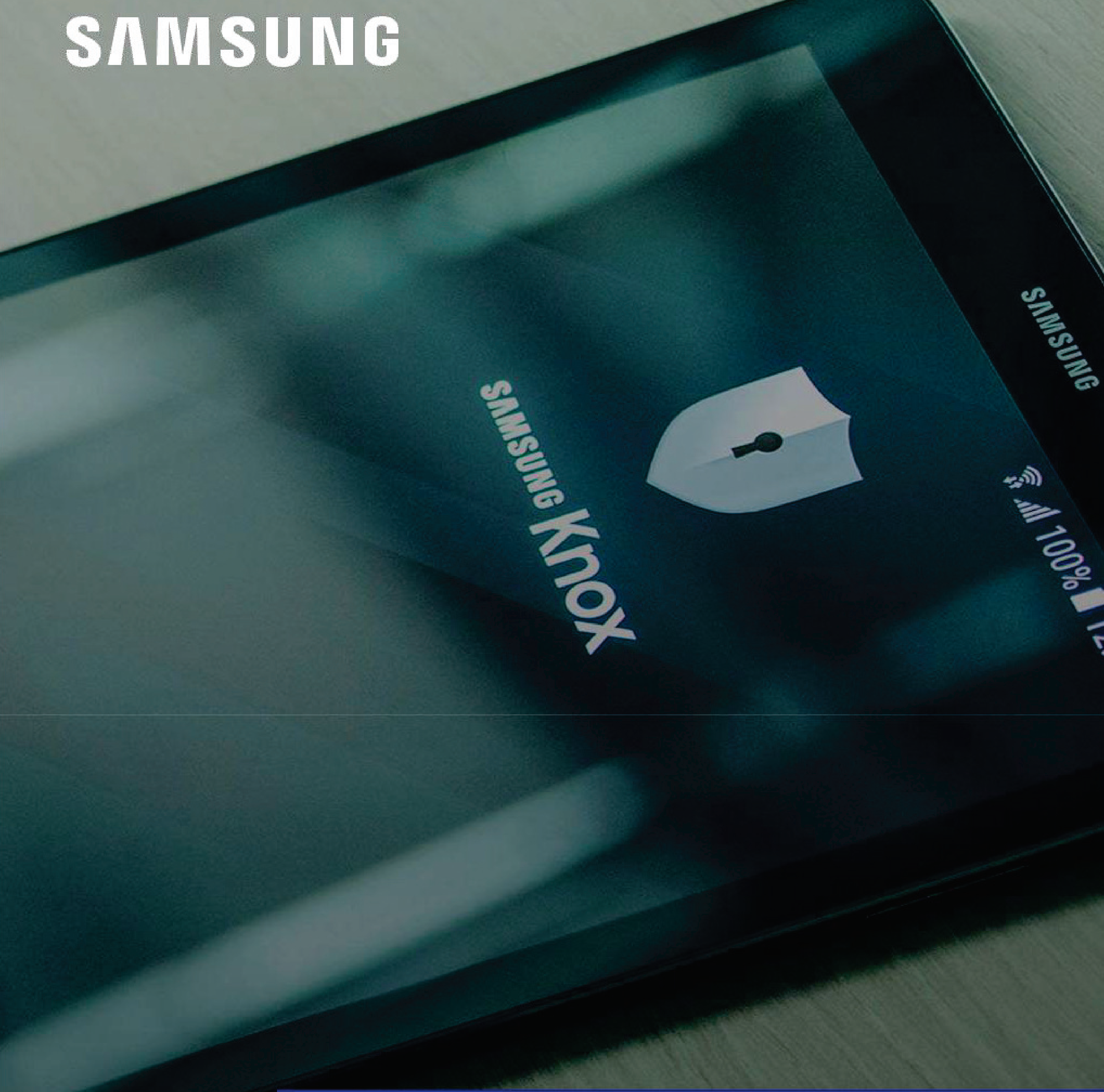
For more information

For more information about Samsung Enterprise Mobility and Samsung KNOX,
visit: www.samsung.com/enterprise and www.samsung.com/knox

SAMSUNG
BUSINESS

EXHIBIT 5

SAMSUNG



Knox Platform for Enterprise White Paper

About this White Paper

This White Paper provides an overview of the Samsung Knox Platform for Enterprise, also known as KPE or Knox Platform, focusing on the unique advantages that differentiate KPE from other options in the mobile device market.

This document is designed for C level executives, security professionals, IT managers, IT admins, and others evaluating KPE as a solution. For additional information about KPE, go to Samsung Knox Product site.

Revision history

Version	Knox Version	Date	Revisions
1.0	3.2 and higher	September 12, 2018	First release.
1.0.1	3.2 and higher	November 1, 2018	Minor revisions.
1.1	3.3 and higher	February 20, 2019	New info about DualDAR Encryption and Knox Verified Boot. Updates to Feature Comparison and Sensitive Data Protection.

Copyright

Contents

Introduction......4
The Samsung Knox Platform......4
Knox Platform differentiators......5
Security highlights......5
Manageability highlights......7
Feature Comparison......8

Core Platform Security......12
Root of Trust......12
Knox Platform trusted environment......12
How the Root of Trust works......13
Hardware-backed security......13
Trusted Boot......15
Secure lockdown on tampering......15
Building on Secure Boot......15
Knox Verified Boot (KVB)......16
Real-time Kernel Protection (RKP)......17
Why does kernel protection matter?......17
RKP design and structure......17
How is kernel protection possible?......18
Full security coverage......19
Device Health Attestation......19
Reliable detection of compromised devices......19
How Knox Attestation works......20
Unique advantages of Knox Attestation......20
Sensitive Data Protection (SDP)......20
Two levels of protection......21
How SDP works......21
SDP protection of apps......22
Unique advantages of Knox SDP......22
App Container......23
Hardware-Backed Security......23
Granular Management Policies......24

Network Security......25
Virtual Private Networks (VPN)......25
Unique advantages of Knox VPN framework......25
Robust handling of enterprise requirements......26
High-security built-in VPN client......26
Network Platform Analytics (NPA)......27
NPA design......28
Unique advantages of Knox NPA......28
NPA-compatible solutions......28

Certificate Management......29
Universal Credential Management (UCM)......29
UCM framework......29
Secure storage options......30
UCM whitelists......30
Client Certificate Manager (CCM)......31
Granular certificate and key access control......31
Signing with device-specific certificates......31
Device integrity assurance......31
Keystore integration with other features......32
Certificate Enrollment Protocols (CEP)......32
CEP asymmetric key acquisition......33
CEP operational environment......33

Device Management......34
Device Software Update Management......34
Why manage device software updates?......34
Strict control over device firmware updates......34
Knox control over user updates......35
Granular Device Management......36
Custom boot banner......36
Split billing (Dual APN)......36
Remote admin lock of device......36
Enterprise roaming......37
Granular policies......37
Samsung DeX Management......39
Why use Samsung DeX?......39
Using Knox to customize DeX......40
Unique advantages of Samsung DeX......40
Firewall Management......41
Why manage and customize device firewalls?......41
Granular control of Internet access......41
Log unsafe URL access......41
Remote Control......42
Unique advantages of Knox Remote Control......42
Audit Log......43

User Authentication......44
Biometric authentication......44
Advantages of Knox Biometrics......44

App and Data Protection......46
Enterprise Productivity Apps......46
Samsung Email......46
Samsung Internet Browser......47
Samsung Contacts......48
Advanced App Management......48
Unique advantages of Knox App Management......48
DualDAR Encryption......50
How DualDAR encryption works......51
Unique advantages of Knox DualDAR......52

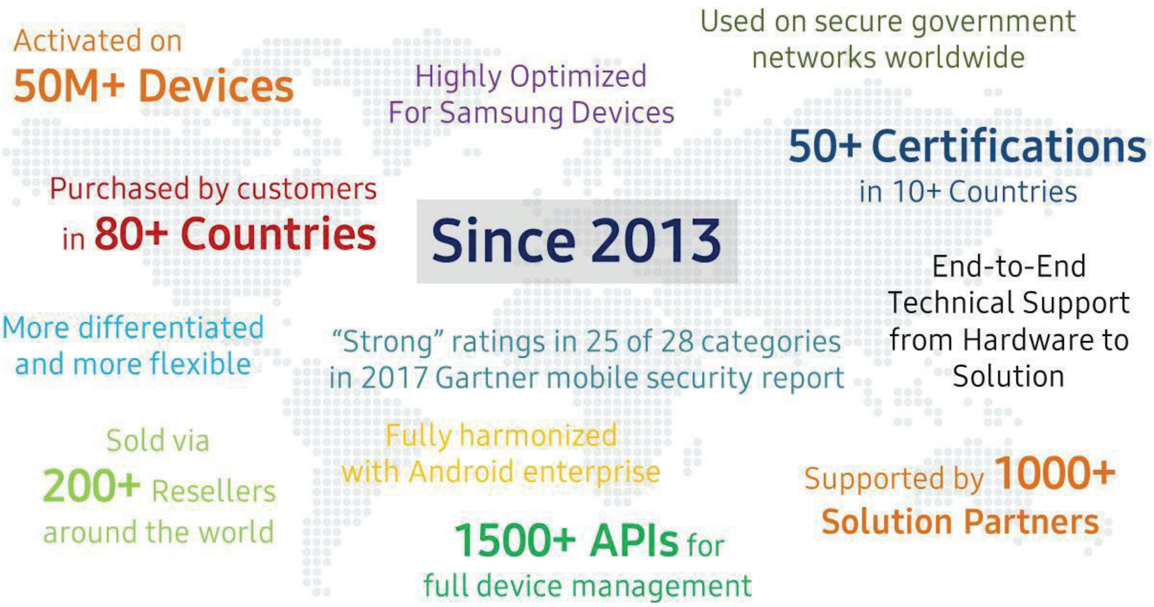
Appendix......53
Knox Certifications......53
Common Criteria Mode......55

Introduction

The Samsung Knox Platform

Samsung’s Knox platform brings defense-grade security on the most popular consumer devices to all enterprises. The Knox Platform provides best-in-class hardware-based security, policy management, and compliance capabilities beyond the standard features commonplace in today's mobile device market. The Knox platform is the cornerstone of a strong mobile security strategy supporting a wide variety of Samsung devices.

Why use the Knox Platform?



The Knox platform helps you and your enterprise avoid the security gaps common on many mobile platforms. Knox received strong ratings in 25 of 28 categories in Gartner’s December 2017 Mobile OSs and Device Security: A Comparison of Platforms and has received strong ratings for the last three years in a row.

The Knox Platform's security hardening supports every aspect of mobile device operation. The Knox Platform enables trust in your mobile endpoints with advanced features like Samsung's patented Real-Time Kernel Protection (RKP) that stands as one of the best kernel protection technologies available from any mobile device vendor. The Knox Platform ensures IT admins can securely bulk deploy the best mobile device hardware, and quickly integrate with existing business infrastructure and apps.

Key benefits for enterprises

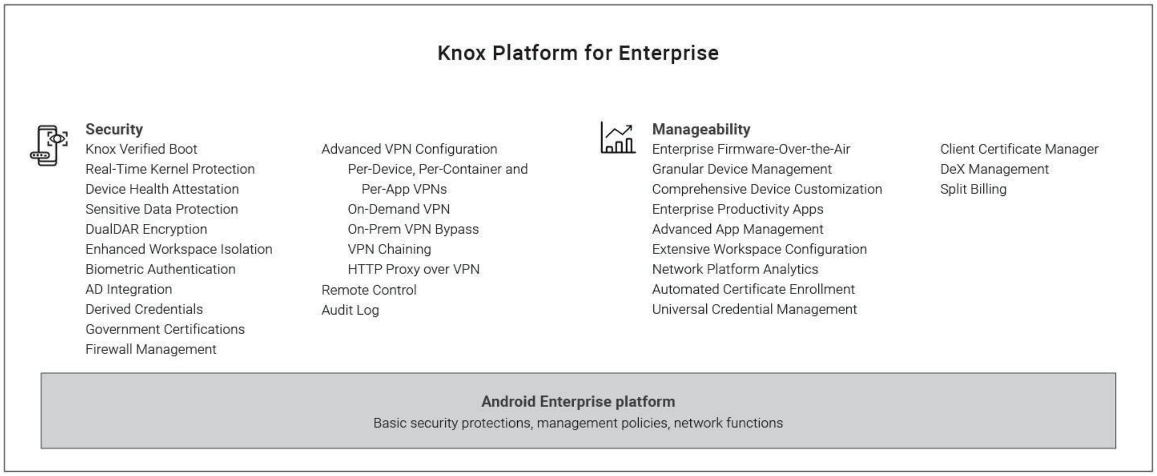
- Easily meet your organization's security and compliance requirements, by providing solid platform integrity, strong data protection, and fine-grained policy enforcement.
- Seamlessly activate and manage Knox Platform features through an Enterprise Mobility Management (EMM) system.

- Flexibly support infrastructure, deployment, and management requirements, through centralized remote device control, advanced VPN management, app whitelisting and blacklisting, and granular policies that control all aspects of Samsung devices.
- Effortlessly upgrade from Android Enterprise, leveraging a comprehensive set of Knox Platform benefits without affecting existing deployments.
- Securely deploy the innovative Samsung Desktop Experience (DeX) in new work environments, unifying mobile and desktop computing on one device.

The Knox Platform's cutting-edge security technology continues to be widely adopted and proven by numerous government, security, and financial agencies throughout the world. Samsung continually works with global government organizations and international regulatory bodies to meet a wide range of certification requirements designed to protect public safety and consumer privacy.

Knox Platform differentiators

The Knox Platform provides a robust set of features that are a superset of features on top of the basic Android platform, to fill security and management gaps, resolve pain points identified by enterprises, and meet the strict requirements of highly regulated industries. The following summarizes the key differentiating features:



Knox Platform for Enterprise

Security
Knox Verified Boot
Real-Time Kernel Protection
Device Health Attestation
Sensitive Data Protection
DualDAR Encryption
Enhanced Workspace Isolation
Biometric Authentication
AD Integration
Derived Credentials
Government Certifications
Firewall Management

Advanced VPN Configuration
Per-Device, Per-Container and Per-App VPNs
On-Demand VPN
On-Prem VPN Bypass
VPN Chaining
HTTP Proxy over VPN
Remote Control
Audit Log

Manageability
Enterprise Firmware-Over-the-Air
Granular Device Management
Comprehensive Device Customization
Enterprise Productivity Apps
Advanced App Management
Extensive Workspace Configuration
Network Platform Analytics
Automated Certificate Enrollment
Universal Credential Management

Client Certificate Manager
DeX Management
Split Billing

Android Enterprise platform
Basic security protections, management policies, network functions

For a quick overview of how these features compare across different platforms, see **"Feature Comparison"** on page **8.**

Security highlights

The following sections describe how the Knox Platform provides an industry-leading ecosystem of products and services to secure and ease mobile device management.

Hardware-backed security

The Knox Platform defends against security threats and protects enterprise data through layers of security built on top of a hardware-backed trusted environment.

- **Trusted environment** — A trusted environment separates security-critical code from the rest of the operating system. This strategic separation ensures only trusted processes that are isolated and protected from attacks and exploits can perform sensitive operations, such as data encryption and decryption. Trusted environments perform integrity checks prior to executing any software. These checks detect malicious attempts to modify the trusted environment and the software running on the device.

- **Hardware-backed** — A trusted environment is hardware-backed if hardware protections isolate the environment from the rest of the running system. This isolation ensures that vulnerabilities in the main operating system don't directly affect the security of the trusted environment. The environment also ties integrity checks of the software running in the trusted environment to cryptographic signatures stored in the device hardware. Hardware-backed integrity checks prevent an attacker from exploiting software vulnerabilities to bypass protections and load unapproved software into the trusted environment.

The Knox Platform uses a hardware-backed trusted environment and the specific components depend on the device hardware. For example, ARM processors provide a Trusted Execution Environment (TEE) that leverages components such as the ARM TrustZone, ARM Hypervisor Mode, and Embedded Secure Elements. Knox features that use the trusted environment include Real-time Kernel Protection (RKP), Trusted Boot, Device Health Attestation, Certificate Management, Sensitive Data Protection (SDP), and Network Platform Analytics (NPA).

App isolation

The Knox Platform uses app isolation to prevent rogue apps from intentionally or inadvertently accessing unauthorized data. The Knox Platform provides several forms of app isolation to create a protected app container space on Samsung devices. Each option is based on the same core isolation technology called Security Enhancements for Android (SE for Android.) SE for Android is an integration of SELinux and Android, expanded to cover Android components and design paradigms. The Knox Platform offers these options:

- **Android Enterprise on Samsung devices** — Android Enterprise provides app isolation through Work Profiles, which provide basic isolation of enterprise apps from personal apps. When using Android Enterprise on Samsung devices, Knox provides features like Real-time Kernel Protection (RKP), secure enterprise apps, and hardware-backed storage of certificates and keys, making Android Enterprise even better on Samsung devices.
- **Knox Workspace** — The Knox Workspace builds on Android Enterprise by providing additional security and management enhancements. Specifically, the Knox Workspace benefits from hardware-backed integrity checks. These checks detect any tampering of the device or its security protections and lock down the Knox Workspace to protect confidential data. The Knox Workspace also supports Sensitive Data Protection (SDP), encrypting data during device runtime and decrypting only after the device user authenticates to unlock the Knox Workspace. Furthermore, the Knox Workspace provides more granular device management, for example, forced two-factor authentication for the Knox Workspace, the use of enterprise Active Directory credentials for authentication, and managed import and export of enterprise data in the Knox Workspace.
- **SE for Android Management Service** (SEAMS) — With SEAMS, you can isolate a single app or small set of trusted apps, to lock down the apps in the same container. App containers created with SEAMS provide the same benefits of the Knox Workspace. Unlike the first two options, however, SEAMS containers have no special GUI. Apps in a SEAMS container appear with the rest of the apps on the device, but are differentiated with a shield badge to show that they're isolated and

protected from apps not sharing their same container. You can create as many of these SEAMS containers as you want on-the-fly.

With the Knox Workspace, enterprises can deploy additional security and management policies to enforce requirements, such as those needed to work within highly regulated industries such as finance, healthcare, and government.

Data protection

Enterprises can protect personal and enterprise data on mobile devices using a rich set of Knox features:

- **User authentication** — Samsung Knox devices support not just password, PIN, and pattern authentication but also the latest biometric authentication: fingerprints, iris, face, and Intelligent Scan. Options are available for both device lockscreen authentication and separate Knox Workspace authentication. Through the Knox Platform, you can provide enforce two-factor authentication for the Knox Workspace or enterprise AD credentials to ensure stronger data protection.
- **Encryption of device data** — Samsung Knox devices provide data encryption through Sensitive Data Protection that binds to the hardware-backed Root of Trust and user authentication. This encryption ensures data is decrypted only on the device where the data is stored, and only by the device owner. DualDAR Encryption offers two instances of encryption to achieve an even higher level of reliability.
- **Encryption of network data** — Samsung Knox devices offer the widest selection of advanced VPN features, providing the ability to configure a separate VPN for the Knox Workspace as well as for individual apps, to reinforce data isolation even further. Knox also offers always-on VPN, on-demand VPN, on-premise VPN bypass, HTTP proxy over VPN, multiple active tunnels, strict data leakage controls, and VPN chaining or cascading.
- **Device tracking, locking, and erasing** — Samsung Knox devices offer the ability to track, geofence, and automatically lock devices based on events and security policies. For example, a device that leaves a specified geographic perimeter is locked, wiped of data, or reset to factory defaults.

Manageability highlights

Device management and deployment

Enterprises with tens, hundreds, or thousands of employee mobile devices need to manage them easily, securely, and efficiently. Through EMM systems, enterprise IT admins can use a web console to centrally and remotely manage devices over-the-air. IT admins can control Samsung Knox devices comprehensively, managing device features with ease.

This management is possible through the Samsung Knox SDK, which offers over 1500 APIs for granular and flexible control over Samsung devices. This functionality is on top of the basic APIs offered through the Android SDK, providing a more powerful superset of capabilities. An EMM app on an employee device receives IT admin commands from the EMM web console, and calls Knox APIs to deploy commands on Knox devices. This integration enables enterprise IT admins to deploy IT policies to manage and secure every aspect of Knox devices.

Device management services

To address a variety of business needs beyond security, the Samsung Knox portfolio is complemented by robust cloud services that ease mobile device deployment, customization, and management. These services include:

- **Knox Mobile Enrollment** — With this free service, enterprises can use a web console or REST API calls to automate device enrollment, either individually or in bulk. After an IT admin registers a device with this service, the device user simply turns it on and connects it to a Wi-Fi or 3G/4G network to enroll it with an EMM system. There is no manual enrollment of individual devices, and no need for IMEI management and verification – all onerous, time-consuming, and error-prone tasks.
- **Knox Configure** — Samsung phones, tablets, and wearables are fully customizable to work in numerous vertical markets such as hospitality, retail, and entertainment. Through a web console, Systems Integrators can create purpose-built devices that present a customized user interface, for example, an information kiosk, point-of-sales terminal, or in-flight entertainment system. The Systems Integrators can customize or restrict almost all aspects of device configuration and the user experience, including boot animations incorporating custom enterprise logos, display settings, wallpapers, network configurations, notifications, and software updates.

Learn more

This White Paper provides an overview of Knox Platform's security features and how they can help resolve common enterprise mobile deployment issues. For information about other Knox features, see the Samsung Knox website.

Feature Comparison

The following table summarizes the advantages provided out-of-box by Samsung Knox devices over non-Samsung devices, and how **Knox Platform for Enterprise** (KPE) extends **Android Enterprise** (AE). For more information, go to the Knox Platform for Enterprise home page.

Feature	AE on non-Samsung devices	KPE Standard on Samsung devices	KPE Premium on Samsung devices	How KPE extends AE
All Android Enterprise Features	●	●	●	KPE extends AE by providing advanced security and manageability controls.
Security				
Secure Lockdown on Tampering	◐	●	●	Upon detecting critical security compromises, the system locks down sensitive areas, preventing enterprise data from being accessed and leaked. In such circumstances, AE restricts access to previously installed keys. KPE extends AE by preventing whole components from running when tampered.
Remote Device Health	◐	●	●	Obtain visibility into which particular devices are experiencing security issues, such as unauthorized firmware, allowing you to

Feature	AE on non-Samsung devices	KPE Standard on Samsung devices	KPE Premium on Samsung devices	How KPE extends AE
				troubleshoot the issue immediately. AE provides software-based SafetyNet APIs, and KPE extends AE by providing reliable hardware-based device attestation.
Knox Verified Boot	◕	◕	●	Knox Verified Boot extends Android Verified Boot by verifying integrity before the device is booted and running, validating the bootloader, TrustZone, and Hypervisor, as well as the kernel.
Audit Log	◕	◕	●	KPE audit log provides comprehensive device logs for troubleshooting potential issues and captures events needed to satisfy government compliance requirements.
Real-Time Kernel Protection (RKP)		●	●	KPE extends AE by providing best-in-class kernel attack prevention features, including kernel code, kernel data, and kernel control flow protections. RKP drastically limits the number of possible attack types against Samsung devices.
Sensitive Data Protection (SDP)		●	●	With basic AE, device data is decrypted once the device boots. With KPE's SDP, selected files remain encrypted at runtime and are decrypted only after a device user authenticates their identity at the device lockscreen, or Knox Workspace login. KPE evicts decryption keys when the device or Knox Workspace locks, and complies with MDFPP requirements for US government and military.
DualDAR Encryption			●	With a single instance of encryption, potential flaws in the implementation can result in a single point of failure. KPE DualDAR provides two independent layers of encryption to achieve an even higher level of reliability by enabling redundancies in protecting Data-At-Rest. This dual encryption is required for classified deployments.
Enforced Two-Factor Authentication			●	KPE extends AE by enabling IT admins to force end-user two-factor authentication for logging into a Knox Workspace, or Managed Device. Authentication can accomplished either using biometrics (fingerprint, iris, face), or with traditional methods (password, PIN, pattern).
Government-Grade Common Criteria Mode			●	KPE extends AE's device controls by exposing a Common Criteria mode to simplify the process of configuring devices into a compliant state for defense deployments.
App Isolation Groups (SEAMS)			●	Unlike classic app containers utilizing a GUI, KPE extends AE by allowing you to manage "invisible" app isolation groups to protect a set of apps from any other set. Up to 300 groupings are possible.
Secure Certificate Enrollment Agents			●	KPE extends AE's certificate management APIs by providing a certificate enrollment service API that closely follow the latest security protocols. There is no reason to enroll certificates insecurely, or implement your own protocols.

Feature	AE on non-Samsung devices	KPE Standard on Samsung devices	KPE Premium on Samsung devices	How KPE extends AE
Manageability				
Manage Device Software Updates	◐	◐	●	A Samsung E-FOTA license enables the controlled rollout of firmware updates upon completion of internal testing, helps avoid compatibility problems with proprietary systems or apps, and minimizes user interaction requirements for updates. KPE provides granular firmware controls that AE does not have. For example, the ability to set highest accepted firmware version, apply specific firmware version to a set of devices at a specific date/time, and the option to block automatic firmware updates.
Remote Control	◐	●	●	KPE extends base remote control capability to allow IT to remotely control employee devices to troubleshoot and fix mobile devices in the field.
Customization	◐	◕	●	With KPE, IT admins can customize various aspects of the device software and UI beyond what is available in AE. Enable/disable task manager, hardware keys, multi-window mode, etc. Custom boot banner/animation, block specific system notifications, customize items appearing on the power off dialog screen, map volume keys to app task switching, and more.
Granular Roaming Controls	◐	◕	●	With KPE, IT admins can allow/disallow the use of "roaming" mobile connections that often incur high call/text/data rates. AE supports disabling mobile data. KPE extends AE by providing additional controls, such as the blocking of calls, or the blocking of app update downloads while allowing other data use. KPE Premium also enables separate roaming controls for each APN to support split billing.
Admin Device Lock	◐	◐	●	An admin device lock enables IT to lock out a device, preventing even valid credentials from being used. This is extremely valuable for managing end-user policy violations, including the travel to hostile countries. While AE supports locking the device screen, it does not lock out the user. KPE extends AE by allowing an IT admin to enforce an admin lock.
Firewall Management		●	●	KPE extends AE by providing an industry-exclusive ability to set device firewall rules. Using KPE, admins can also be notified when employees attempt to visit blocked domains.
Granular Device Policies		◐	●	With KPE's granular device policies, an enterprise can meet compliance or other deployment requirements using unique policies, that are not supported on AE, for SMS/MMS disclaimers, call restrictions, read and write restrictions on SD cards, granular Bluetooth profile restrictions. KPE's refined device policies can even manage DeX deployment settings.

Feature	AE on non-Samsung devices	KPE Standard on Samsung devices	KPE Premium on Samsung devices	How KPE extends AE
Advanced Workspace Configuration			●	KPE extends AE by providing container-specific policy settings. KPE enables strict policy enforcement for Bluetooth, SD Card, USB, and other technologies inside the container, while allowing the full use of these technologies outside the container.
Unlock using Active Directory Credentials			●	With KPE, there is no requirement for employees to remember separate credentials for Windows laptops and mobile devices. Additionally, with KPE device users can utilize their existing Active Directory credentials to unlock their devices.
Split Billing (Dual APNs)			●	KPE extends AE through the support of dual APN management. KPE enables enterprises to pay only for the data usage of approved business apps. Employees are then responsible for fees incurred for personal data usage.
Network Analytics			●	KPE allows an IT admin to deploy network threat detection solutions without granting tools complete access to all network traffic.
VPN				
VPN Granularity: Per-App, Per-Container, or Whole Device	◐	◐	●	KPE extends AE to provide very granular VPN controls. KPE can be configured with a VPN tunnel not just for a container or individual apps, but for the whole device.
Always On VPN	◐	◐	●	KPE utilizes strict controls to block traffic from bypassing a configured VPN, even in cases where the VPN client crashes or when the device is rebooting. AE does not block traffic when a VPN is down.
On-Demand VPN			●	A KPE VPN can be set to only activate when certain target apps are launched/running, and does not require additional VPN client support.
HTTP Proxy over VPN			●	A KPE VPN enables the use of web proxies on tunneled VPN traffic.
VPN Chaining			●	A KPE VPN allows the use of two VPN tunnels to double-encrypt traffic, enhance anonymity, and prevent a single security bug in a VPN layer from compromising network encryption.
Near-instant VPN connection times		●	●	The Knox VPN framework allows a near-instant VPN connection, clocking in at one second. This time is measured from when the VPN handshake and authentication completes, to when the tunnel is established and traffic from any tunneled apps can pass through the VPN. This time threshold applies to all apps, assuming 100 apps enrolled in the VPN profile, whether they are part of the Knox Workspace or not.

Core Platform Security

Root of Trust

Imagine every device in your network simultaneously infected with malware and combing through your confidential data. Attacks and exploits continue to mature in sophistication in an attempt to stay ahead of advancing mobile device safeguards. So what's the single solution that works on all devices at the same time? To build a robust Root of Trust stack that minimizes exposure, detects intrusions, and locks down sensitive information.

A Root of Trust is the cornerstone of any modern security protocol. It is a series of stringent checks and balances, beginning at the hardware level rather than the software level. This feature adds a level of security to devices, making them difficult to subvert as hardware is more immutable than software.

A Root of Trust answers many complicated security questions, such as:

- How do you **know** if a compromised OS was booted at runtime?
- Can you **trust** that your certificates are stored securely?
- Has an exploit **modified** the kernel or other system software?

Samsung’s approach to addressing this issue is to bottleneck all security-critical functionality through trustworthy components. These trustworthy components are thoroughly designed, reviewed, and maintained with the following considerations:

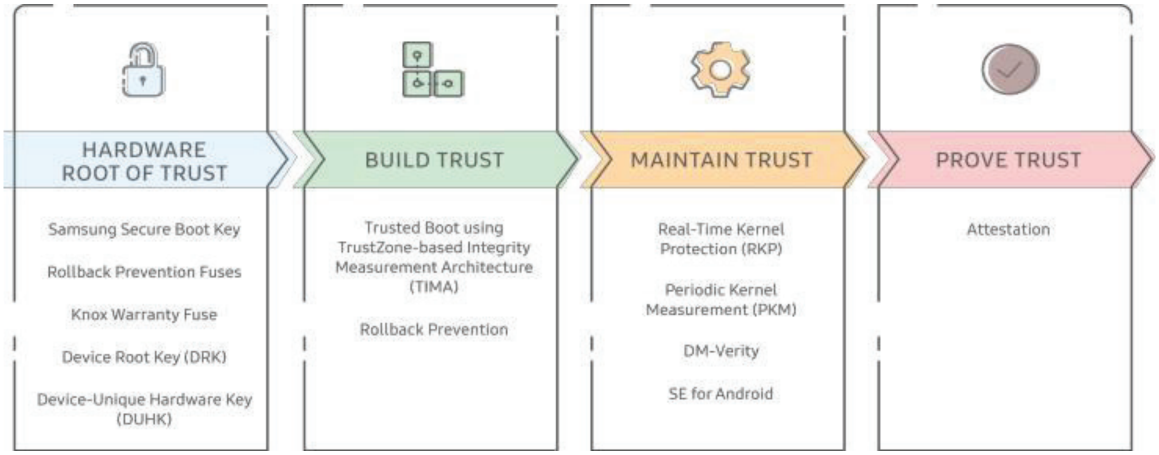
- **What are the assurances required?** High-security enterprise partners require a near-total ability to control and audit the software that interfaces with their systems. End users must have the authority to deny permission to use their device features and data. Each user, partner, and integrated system has its own requirements, many of which are assured in large part through the Roots of Trust.
- **How can components contribute to more complex assurances?** A Trusted Boot process enables the trustworthy transfer of control from the bootloader to the Android framework. This trustworthy transfer of control plays a key role in the admin's ability to audit apps running on the device.
  Secure boot is a complex process built on top of many smaller components that validate software, configuration files, deployment processes, and update processes. Each of these smaller components contributes to the secure boot process, and a secure boot process itself contributes to the security of other processes.
- **How can we make these components, their assurances, and their usage more robust?** Each Trusted Application on a Samsung Knox device ultimately represents a Root of Trust. These Trusted Applications encompass functionality such as device identity, key management, and remote attestation of device health. Samsung Knox uses these same Trusted Applications to provide its own assurances.

Knox Platform trusted environment

The Knox Platform builds a unique, industry-leading trusted environment in four ways:

- **Establishes** a hardware-backed Root of Trust, on which other components rely.
- **Builds** trust during boot, through features like Trusted Boot.
- **Maintains** trust while the device is in use, through features like Real-Time Kernel Protection.
- **Proves** its trustworthiness on demand, through Device Health Attestation.

This process and its components are as follows:


HARDWARE ROOT OF TRUST
Samsung Secure Boot Key
Rollback Prevention Fuses
Knox Warranty Fuse
Device Root Key (DRK)
Device-Unique Hardware Key (DUHK)
BUILD TRUST
Trusted Boot using TrustZone-based Integrity Measurement Architecture (TIMA)
Rollback Prevention
MAINTAIN TRUST
Real-Time Kernel Protection (RKP)
Periodic Kernel Measurement (PKM)
DM-Verity
SE for Android
PROVE TRUST
Attestation

How the Root of Trust works

1. Knox Platform security starts in the factory—before users even power on their mobile device— when a Device-Unique Hardware Key (DUHK) is generated on the device using its hardware random number generator.
2. Next, the DUHK generates and encrypts the Device Root Key (DRK) and Samsung Attestation Key (SAK). The DRK and SAK contain an authentication code that enables recipients to verify the IMEI and serial number of provisioned devices. Since existing purchasing systems use a device's IMEI and serial number to track devices and not the DRK identifier, this enables users to obtain proof they are interacting with devices they have purchased. The use of the DUHK is only available to the TrustZone operating system. The TrustZone OS uses the DUHK to create subsequent keys unique to each trusted application. Trusted applications uses these keys to securely store data. The DRK and SAK are private keys that enable trusted applications to prove their own identity, as well as the identity of the device they are executing on. These trusted applications integrate deeply with hardware to provide hardware-backed security.
3. Upon device start up, Samsung uses the Samsung Secure Boot Key (SSBK) to check all software components. One of the components is the TrustZone Secure World, a chip partition reserved for secure code and data. Only specially privileged software modules running within the TrustZone Secure World can access these keys.
4. The software performs a check on each Knox Platform feature before allowing it to run. Since this chain of security checks begins with the very first hardware check, each feature is protected by hardware Root of Trust. No matter which link in the chain an attacker targets, one of the security checks detects it.

Hardware-backed security

The Knox Platform trusted environment leverages the following hardware components:

Secure hardware

- **ARM TrustZone Secure World** — The Secure World is the environment where highly sensitive software runs. The ARM TrustZone hardware ensures memory and components marked secure (for example, a fingerprint reader) can only be accessed in the Secure World. Most of the system, including the kernel, middleware, and apps, run in the Normal World. The Secure World software, on the other hand, is more privileged, and can access both Secure and Normal World resources.
- **Bootloader ROM** — The Primary Bootloader (PBL) is the first piece of code to run during the boot process. The PBL is trusted to measure and verify the boot chain. To prevent tampering, the PBL is kept in the ROM of the secure hardware. The device hardware loads and runs the PBL from ROM at boot, and the PBL starts the Secure and Trusted Boot processes.

Hardware keys

- **Device-Unique Hardware Key (DUHK)** — Samsung incorporates the DUHK, a device-unique symmetric key, in device hardware during the initial manufacture of the device. The DUHK binds data—for example, device health attestation data— to a particular device and is accessible only to a hardware cryptography module and not directly exposed to any device software. However, software can request that the DUHK encrypt and decrypt data. This DUHK encrypted data is bound to the device, and cannot be decrypted on any other device.
- **Device Root Key (DRK)** — The DRK is a device-unique, asymmetric RSA key pair that is signed by Samsung's root key through an X.509 certificate. This certificate proves that Samsung produced the DRK. The DRK is generated at manufacture in the Samsung factory and is stored on the device encrypted by the DUHK, thus binding it to the device. The DRK is only accessible from within the TrustZone Secure World and is protected by the DUHK. The DRK is an important part of the Root of Trust, as it derives other signing keys. Because the DRK is device-unique, it can tie data to a device through cryptographic signatures. Signing keys are derived from the DRK and used to sign data.
- **Samsung Secure Boot Key (SSBK)** — The SSBK is an asymmetric key pair used to sign Samsung-approved boot executables.
  - The private part of the SSBK is used by Samsung to sign secondary and app bootloaders.
  - The public part of the SSBK is stored in the hardware's one-time programmable fuse at manufacture in the Samsung factory. The Secure Boot process uses this public key to verify whether each boot component it loads is approved.
- **Samsung Attestation Key (SAK)** — The SAK is also a device-unique, asymmetric key pair that is signed by Samsung's root key. This signed key pair proves that the SAK was produced by Samsung. The SAK is used to sign the Attestation blob that indicates if the device is in a trusted state. The signature proves that Attestation data originated from the TrustZone Secure World on a Samsung device. Unlike the DRK, the SAK is a set of ECDSA keys. ECDSA is a newer asymmetric algorithm, similar to RSA but smaller and faster for the same strength.

Hardware fuses

- **Rollback Prevention (RP) fuses** — RP fuses encode the minimum acceptable version of Samsung-approved bootloaders. Old software may contain known vulnerabilities that may be exploited. Rollback prevention excludes approved, but out-of-date bootloaders from being loaded. The RP fuse version number is set when system software is initially installed and when specific updates occur. Once the RP fuse version number is set, it is impossible to revert back to legacy software versions.

- **Warranty fuse** — The Knox Platform uses a one-time programmable fuse that signifies whether or not the device has ever booted into an unapproved state. If the Trusted Boot process detects non-approved components are used, or if certain critical security features such as SELinux are disabled, it sets the fuse. When the fuse is set, the following security measures take place:
  - Device Health Attestation checks fail.
  - The Knox Keystore removes the keys used by Sensitive Data Protection for data encryption and decryption, preventing access to sensitive data.
  - The Knox Workspace no longer operates, preventing access to secured enterprise apps and the data within.

Trusted Boot

Trusted Boot is a Knox Platform feature representative of Samsung's industry leading mobile device boot protection. Trusted Boot identifies and distinguishes unauthorized and out-of-date boot loaders before they compromise your mobile device.

If unauthorized boot components happen to load, an enterprise can trust that only validated and current components are loaded after Trusted Boot segregates authorized from unauthorized boot loaders.

Enterprises can check device integrity on demand through Knox Attestation, which reads Trusted Boot collected measurement data, along with an SE for Android enforcement setting, to form the basis of a device health verdict.

Secure lockdown on tampering

Bootloader measurements are recorded in secure TrustZone memory during device boot. At runtime, apps operating in the secure TrustZone can use these measurements to make security-critical decisions, such as whether or not to:

- Release cryptographic keys from the Knox Keystore.
- Launch the Knox Workspace app container.

If an unauthorized or out-of-date component version is detected, a tamper fuse is set. Once the fuse is set, sensitive work apps and data within the Knox Workspace are permanently encrypted and inaccessible since the integrity of the device is no longer guaranteed or validated.

The device user can still boot the device and launch personal apps. This flexibility promotes a nice balance between consumer functions, such as smartphone calls and personal apps, and the requirement to protect enterprise data.

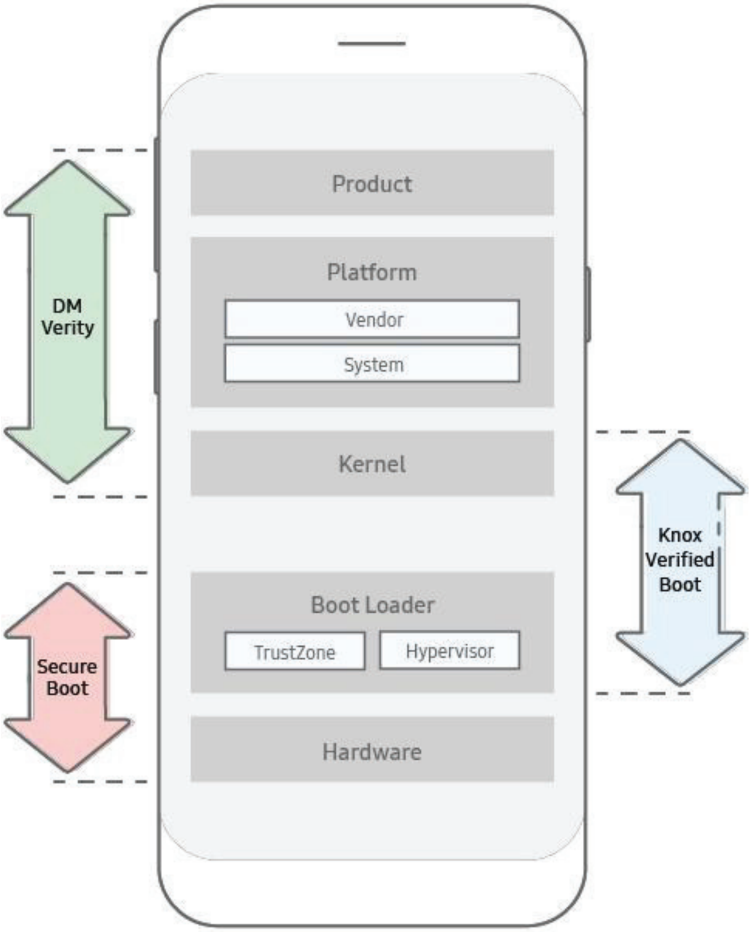
Building on Secure Boot

Before adopting Trusted Boot to work along with Secure Boot, Samsung devices were using Secure Boot to prevent unauthorized bootloaders and operating systems from loading during start-up. Secure Boot is implemented by each bootloader cryptographically verifying the signature of the next bootloader in sequence, using a certificate chain with its root-of-trust resident in hardware. If verification fails at any step, the boot process terminates.

While Secure Boot is effective at preventing unauthorized bootloaders, it is unable to distinguish between different authorized binary versions. For example, Secure Boot can't distinguish between a bootloader with a known vulnerability as opposed to a later patched version, since both versions have valid signatures. Trusted Boot however was introduced to verify the same bootloader, kernel and platform build.

Knox Verified Boot (KVB)

Knox Verified Boot (KVB) is a new solution that both extends and enhances Android Verified Boot (AVB). While AVB only checks the integrity of the kernel and platform components, KVB extends those checks to also cover the earlier bootloaders. This provides a more comprehensive guarantee the device is booting using properly signed components that are all from the same build. KVB performs the same type of validations as the existing Trusted Boot mechanism, but it is able to do so before the device kernel is booted, and thus provides the same data protection guarantees earlier.



With KVB, component checks are conducted in the bootloader, and validations are made before system services are even started.

KVB is supported on Samsung S10 and above devices running the Android P operating system or later.

Real-time Kernel Protection (RKP)

The Knox Platform's patented Real-time Kernel Protection (RKP) is the industry's strongest protection against kernel threats and exploits. RKP works seamlessly out-of-the-box, with no setup required. Simply powering on a Samsung Knox device provides world-class threat protection and attack mitigation. RKP supports the rest of the Knox security offerings to provide full security coverage without the typical gaps anticipated with mobile devices.

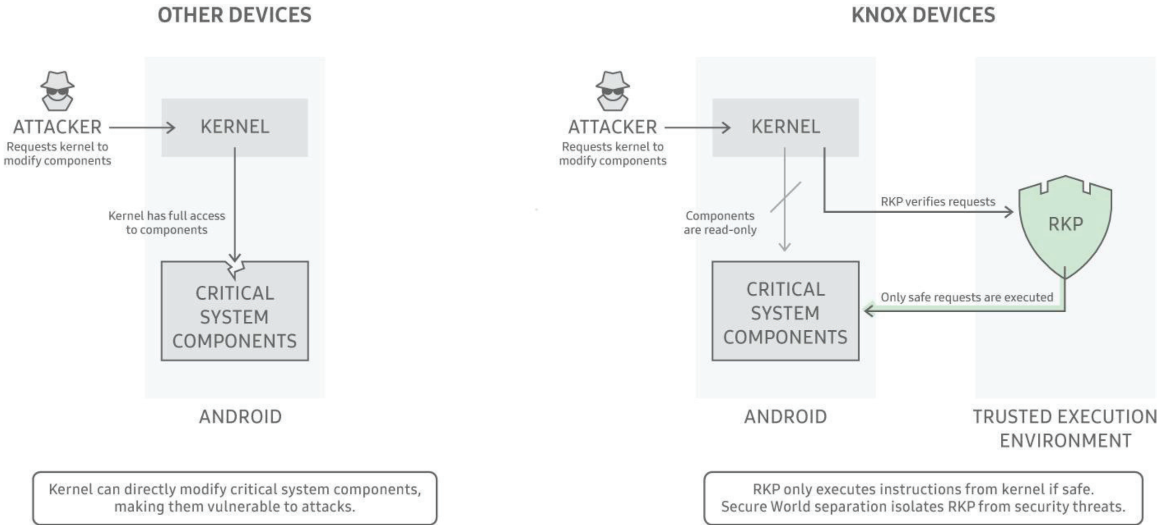
Why does kernel protection matter?

Kernel protection is central to device security and enterprise data protection. When attackers find software vulnerabilities, they often escalate privileges and compromise the core of the OS: the kernel.

A compromised kernel can leak sensitive data and even allow remote monitoring and control of the affected device. Other more commonplace protections like Secure Boot or hardware-backed keystores are of little value if the kernel itself is controlled at runtime. After a device boots and decrypts sensitive content, a kernel compromise can result in data leaks that directly impact an enterprise's data integrity.

RKP design and structure

As part of the Knox Platform's security offerings, RKP employs a security monitor within an isolated execution environment. Depending on the device model, either a dedicated hypervisor or the hardware-backed secure world provided by ARM TrustZone technology provides the isolated execution environment.


OTHER DEVICES
ATTACKER
Requests kernel to modify components
KERNEL
Kernel has full access to components
CRITICAL SYSTEM COMPONENTS
ANDROID
Kernel can directly modify critical system components, making them vulnerable to attacks.
KNOX DEVICES
ATTACKER
Requests kernel to modify components
KERNEL
Components are read-only
RKP verifies requests
RKP
Only safe requests are executed
CRITICAL SYSTEM COMPONENTS
ANDROID
TRUSTED EXECUTION ENVIRONMENT
RKP only executes instructions from kernel if safe. Secure World separation isolates RKP from security threats.

RKP's isolation from the kernel shrinks the Trusted Computing Base (TCB) and helps secure it from attacks designed to compromise the kernel. This unique ability enables RKP to detect and prevent the most common kernel attacks. RKP protections are grouped into three areas:

- **Kernel code** — RKP prevents modification of kernel code and logic.
- **Kernel data** — RKP prevents modification of critical kernel data structures.

- **Kernel control flow** — RKP prevents Return-Oriented Programming (ROP) and Jump-Oriented Programming (JOP) attacks that reuse existing kernel logic to piece together exploits from the kernel's own code.

How is kernel protection possible?

A kernel protection mechanism can't exist completely in the kernel only, since an attacker could circumvent it if the kernel itself has a flaw. The kernel is the lowest granular control level over the OS and, as such, usually can't be effectively monitored from any lower level in the system.

RKP uniquely employs a security monitor within an isolated execution environment. Running within an isolated execution environment would normally compromise a security mechanism's ability to see into the kernel and monitor activities at runtime. However, RKP succeeds by utilizing patented techniques to control device memory management and by intercepting and inspecting critical kernel actions before allowing them to execute. RKP is thus able to prevent a compromised kernel from bypassing other security protections. This prevention significantly reduces the severity of kernel attacks and limits the effectiveness of exploits that would typically cripple a mobile device.

Since RKP is always active and requires no management control, kernel protection is only possible if it meets strict usability and performance requirements. RKP's protections are activated out-of-the-box, with no performance impact to customers.

Periodic Kernel Measurement (PKM)

The TrustZone-based Integrity Measurement Architecture (TIMA) architecture provides a number of core features to protect against mobile device compromise. One of these central TIMA features is Periodic Kernel Measurement (PKM).

PKM periodically monitors the kernel to detect if legitimate kernel code and data were modified maliciously. PKM also monitors the key SE for Android data structures in OS kernel memory to prevent malicious attacks from corrupting and potentially disabling SE for Android. PKM protects the Linux kernel code and data pages from malicious exploits and helps prevent attacks attempting to disable SE for Android.

During a device firmware build, the SHA1 hash of every kernel code, and read-only data page, is calculated and gathered into a measurement file. These measurements are signed by Samsung to ensure data integrity and authenticity before its included in the firmware. When TIMA is initialized, PKM receives the kernel page measurements and verifies the signature to prove integrity and authenticity before storing the measurements in the secure world. During device operation, TIMA periodically recalculates the measurements of the running kernel and compares them to the signed measurements stored on the device. If any discrepancy is detected, a violation is reported to both system logs and the user.

When PKM runs, it reads the physical memory addresses used by SE for Android to determine whether:

• SE for Android is enabled

• SE for Android is in enforcing mode.

If malicious code manages to disable SE for Android, or switch it to permissive mode, PKM detects the state change and reports a violation to quickly assist an administrator in problem diagnoses.

Full security coverage

Each year, Samsung's research and development teams add the latest runtime protections to a growing list of unique capabilities found only within RKP.

Although RKP is only one piece of Samsung's holistic security solution, it successfully demonstrates the unique security guarantees possible when combining hardware, software, and advanced security research. Ensuring security claims are low maintenance, highly effective, and industry-leading is what provides enterprise customers the trust they need to deploy mobile devices in high-security environments.

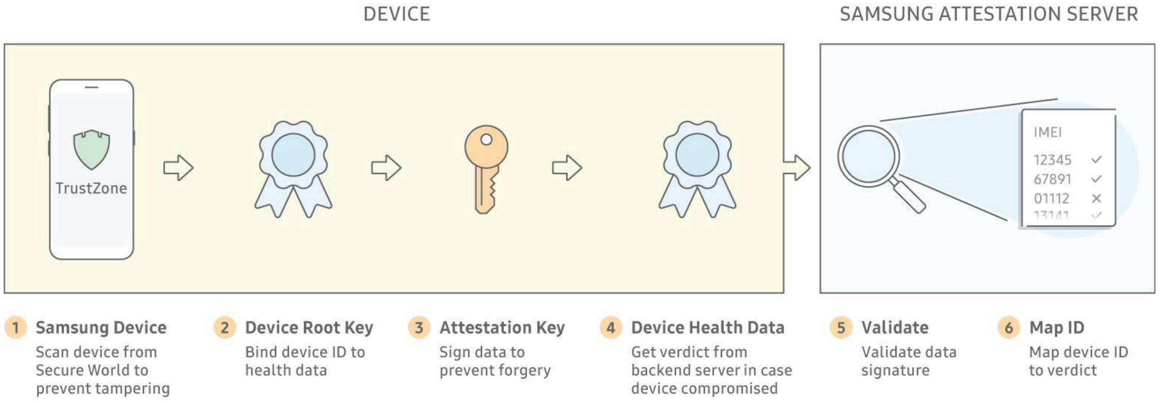
Device Health Attestation

A mobile device can be compromised if unauthorized agents gain super-user access permissions to the powerful system files that control device operation and data access. This loss of control is possible if a device user roots their device to get full control over the device firmware, files, UI, and apps. Unfortunately, malware can exploit this vulnerability to steal passwords, hijack identities, access secret info, install apps, and modify firmware.

Enterprises with Bring Your Own Device programs are especially at risk, as employees may potentially use rooted Android devices in the workplace. Risks range from the undetected exposure of confidential enterprise assets to wider more insidious attacks on other enterprise resources and infrastructure. Enterprises must have a fail-safe way to detect if a device is compromised, before allowing device users to deploy it in the workplace.

Reliable detection of compromised devices

Malware can potentially intercept and forge the results of a device health check, making a compromised device seem secure. The Knox platform leverages its hardware-backed trusted environment to reliably detect and report compromised devices.


DEVICE
SAMSUNG ATTESTATION SERVER
TrustZone
IMEI
12345 ✓
67891 ✓
01112 ✕
13141 ✓
1 Samsung Device
Scan device from Secure World to prevent tampering
2 Device Root Key
Bind device ID to health data
3 Attestation Key
Sign data to prevent forgery
4 Device Health Data
Get verdict from backend server in case device compromised
5 Validate
Validate data signature
6 Map ID
Map device ID to verdict

Because a Device Root Key (DRK) is unique to each device, it can tie data to a device through cryptographic signatures. The Samsung Attestation Key (SAK) signs the Attestation data to prove that it originated from the TrustZone Secure World on a Samsung Knox device.

Knox Attestation works in tandem with Trusted Boot and Periodic Kernel Measurements to ensure the integrity of devices during deployment, bootup, and operation.

How Knox Attestation works

1. A device check is initiated by either:
    - An enterprise IT admin using an EMM console
    - A web script executing a regularly scheduled check
2. The web server that initiated the check requests a nonce from Samsung's Attestation server. A nonce is just an arbitrary number used in cryptographic communication to uniquely identify each attestation result.
3. The web server instructs the device to begin a check, passing the nonce as a check identifier.
4. A Knox Attestation agent on the device operates within the Secure World partition within the ARM TrustZone to create a blob, that is, a binary large object. This blob is a snapshot of the device's current state. It contains data about whether the device was ever rooted, or if the device has a bootloader or firmware file that was not factory installed or part of an official upgrade.
5. Samsung's Attestation server validates the data signature on the blob to ensure that it was from a trusted Samsung source, analyzes the blob data, and derives a verdict indicating whether or not the device is compromised.
6. The original requestor of the device check can quickly take action, for example,
    - Report the verdict to the device user.
    - Immediately prevent the device from accessing enterprise systems.
    - Uninstall any enterprise apps or assets already on the device.

Unique advantages of Knox Attestation

Knox Attestation provides these key differentiators:

- Health measurements guaranteed per device, through a Device Root Key.
- Health results that easily map to device identifiers like an IMEI.

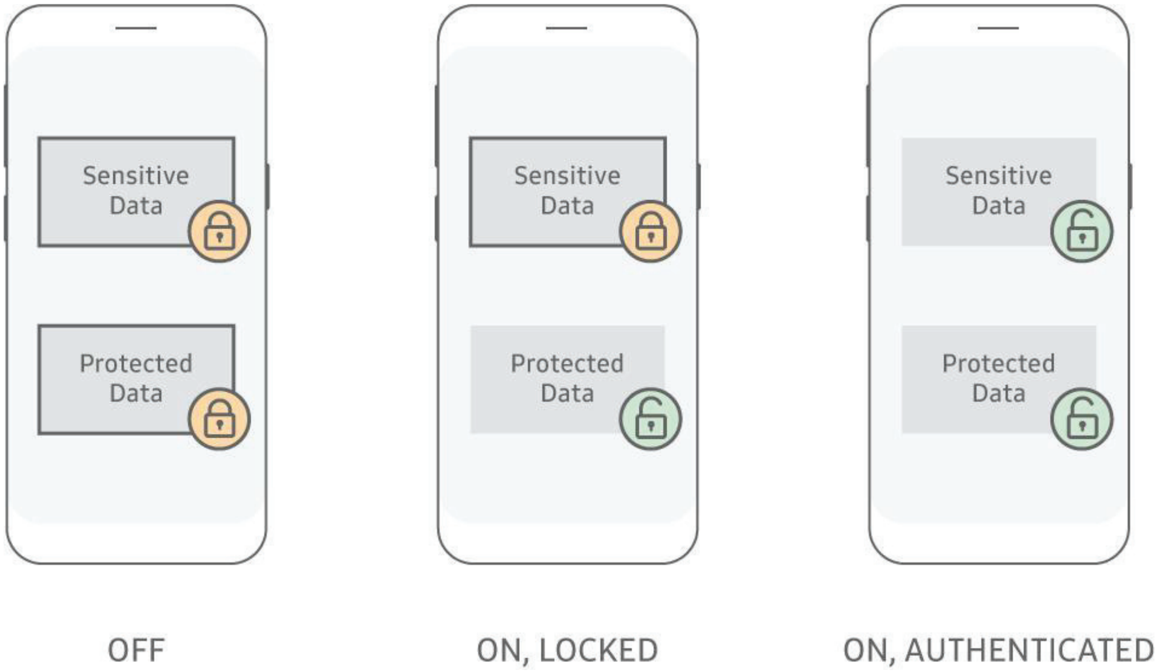
Unlike other solutions on the market, Knox Attestation enables IT admins to determine which attestation result correlates with which device, without having to painstakingly map IDs manually. With competitor solutions, results are returned for separate devices, but IT admins can't differentiate between devices, and consequently the results are not actionable. Knox Attestation returns a single device ID and enables IT admins to prevent or contain issues promptly.

Sensitive Data Protection (SDP)

Protecting Data-At-Rest (DAR) on mobile devices is a major concern. While the industry standard is to encrypt all the data on a device, that data is decrypted and accessible after the device boots successfully. This access process means that once a device is lost or stolen, a sophisticated attack can extract data as long as the device is still running, even if the device is locked. Samsung created Sensitive Data Protection (SDP) to address this specific issue.

SDP meets the Mobile Device Fundamentals Protection Profile (MDFPP) requirements defined by the National Information Assurance Partnership (NIAP) for DAR, meaning that SDP is approved for use by the US government and military.

Two levels of protection



KPE protects user data on the device through Data-at-Rest encryption. Data remains encrypted on disk, and can only be decrypted when the device is powered on. Recovery of data decryption keys is tied to:

- device hardware, meaning data is recoverable only on the same device
- device boot-time integrity measurements
- a user credential dependent on configuration

Additionally, a mechanism is provided to optionally mark data as sensitive, which subsequently cannot be decrypted while the device is in the locked state. Here are the two protection modes that KPE provides for Data-at-Rest:

- **Protected**: All files stored on the device are treated as Protected by default. Protected data is stored on the device file system as encrypted data, and is only decrypted when an application accesses the data. This mechanism provides the data-at-rest protection while the device is powered off. Even if the device is in the lock state, applications can access protected data.
- **Sensitive**: Files can also be optionally marked as sensitive, using the Sensitive Data Protection (SDP) mechanism. SDP uses a key management scheme which ensures sensitive files can only be decrypted in the unlocked state, by purging keys from RAM when the device is locked. However, SDP also provides the ability for new files to be written and encrypted in the locked state using public key cryptography.

How SDP works

Samsung Galaxy devices supporting Knox 3.3 and above are enabled to support Android's File Based Encryption (FBE) for Data-at-Rest. Data encryption is enforced across the device using:

- EXT4 encryption FBE mechanism
- FIPS compliant hardware crypto module (AES256-XTS)

Optionally, the external SD Card can be used with:

- eCryptfs stacked file system
- FIPS compliant Kernel crypto module (AES256-CBC).

FBE keys are derived using a password entry, which is either the default hard-coded password or the device user's password used to unlock the device.

While in the unlocked state, SDP works as follows:

- Encrypts sensitive data using a per-file File Encryption Key (FEK). These keys are encrypted with the SDPK.sym (Sensitive Data Protection Key, symmetric), which is encrypted by the SdpMasterKey.
- Keeps the SdpMasterKey in memory only while the device is unlocked, to allow decryption of the SDPK.sym and SDPK.pri (private).
- Encrypts the SdpMasterKey using the key that is protected by both ephemeral keys derived from the device user's password and a key chaining to the Root Encryption Key (REK) using the Keystore.
- Clears the SdpMasterKey when it transitions to the locked state, and re-derives it when the user unlocks the device or workspace.

While in the locked state, SDP handles apps writes of sensitive data differently:

- Rejects app attempts to open sensitive data files, as KPE no longer has the keys needed to retrieve sensitive data in memory and cannot re-derive them until the user unlocks the device or workspace.
- Encrypts any new sensitive app data by using both a:
  - per-user sensitive data ECDH asymmetric key pair (SDPK.pri/pub)
  - per-file ECDH key pair [DataK.pri/pub] generated on behalf of the app
- Protects the private portion of the ECDH key pair (SPDK.pri) with the SdpMasterKey, the same Key Encryption Key (KEK) used to encrypt the sensitive data per-file FEKs.
- Clears the SdpMasterKey when it transitions to the locked state.

SDP protection of apps

The native **Samsung Email** app automatically uses SDP to protect email bodies and attachments. For performance reasons, the email header (including the subject and sender) is not protected with SDP.

The **Knox Chamber** is a dedicated directory in the Knox container file system. All stored files within the Knox Chamber directory are automatically marked as sensitive and are handled by the SDP mechanism.

Unique advantages of Knox SDP

- **MDFPP-Compliant** — Knox SDP is certified as MDFPP-compliant. Without Knox SDP, the base Android system is not certified as satisfying MDFPP requirements, which mandates a form of SDP.

MDFPP compliance is a requirement for many government agencies and the companies they work with. Samsung has more MDFPP-certified products than any other mobility solution provider.

- **Granular Control** — You can use Knox SDP to protect not just the whole device, a container, or individual files but also selected database columns.
- **Per-App Password** — You can further customize Knox SDP to decrypt a particular app's Sensitive Data only after an app user enters an app-specific password. In this case, the device or container unlock authentication alone does not decrypt app data. An app password is also needed for a higher layer of security.
- **App Protection** — Knox SDP is enabled by default to secure both Samsung Email as well as Knox Chamber.

App Container

Device users typically want their personal data and work data on one device. This requirement presents a challenge for enterprises, which need to ensure that:

- Work data is fully protected; and
- They don't run into any liability issues by accidentally interfering with a user's personal data.

Knox Workspace is an app container that provides enterprises with a solution to securely isolate personal and work data on one device. Protected by best-in-class hardware security, Knox Workspace provides IT admins with granular management policies. The Knox Workspace goes far beyond the standard data isolation provided by competitor container solutions.

Hardware-Backed Security

The Knox Workspace benefits from many of the Knox platform security features. For example:

- Device users can't create or use the Knox Workspace if the device is compromised due to unauthorized boot loaders or unauthorized modifications.
- The device's Knox Workspace data is protected against these types of kernel exploits, which can compromise other mobile platforms, in one or more of the following ways:
  - A malicious process in the personal space exploiting the mapping of kernel data.
  - Privileges of processes running in the personal space escalating to allow access to data in the Workspace.

Granular Management Policies

The Knox Workspace provides an IT admin with granular management policies to address the challenges of maintaining personal and work data on the same device.

Data Transfer

With the isolation of work and personal data, a device user has access to two separate spaces. To increase productivity in certain situations, it is often required to share data from one space to another. For example, while using a phone app in the personal space, it may be necessary to call a work contact saved in the secure work space. With the Knox Workspace, an IT admin has the granular management policies to manage the import and export of data to and from the Knox Workspace. This data can include apps, files, clipboard data, call logs, contacts, calendar events, bookmarks, notifications, shortcuts, and SMS.

Container-Only Control

For liability and productivity purposes, an IT admin can't apply effective policies on a mobile device with both personal and work data. The Knox Workspace provides the IT admin the ability to configure and control critical functionality for the container only. An IT admin can enable or disable the following exclusively for the container:

- Bluetooth
- NFC
- USB access
- External storage

Container Configuration

With the isolation of work and personal data, the device user has access to two separate spaces. This dual access presents some challenges to quickly identify and access work data.

To enhance usability, the Knox Workspace provides an IT admin the ability to add work shortcuts to their personal space so they can quickly access work data. The Knox Workspace also provides an IT admin with the ability to set custom resources, such as work badges on app icons so a user can quickly identify their company’s work apps.

Password Policy

An enterprise IT admin must ensure only authorized people have access to work data inside container. The Knox Workspace supports advanced authentication mechanisms to meet all enterprise needs.

An IT admin can enforce and configure:

- Complex password or code scheme
- Two-factor authentication
- Active Directory authentication

Additionally, an IT admin can lock the container to restrict access. This restriction is necessary when a device is out of compliance, lost, or stolen.

Network Security

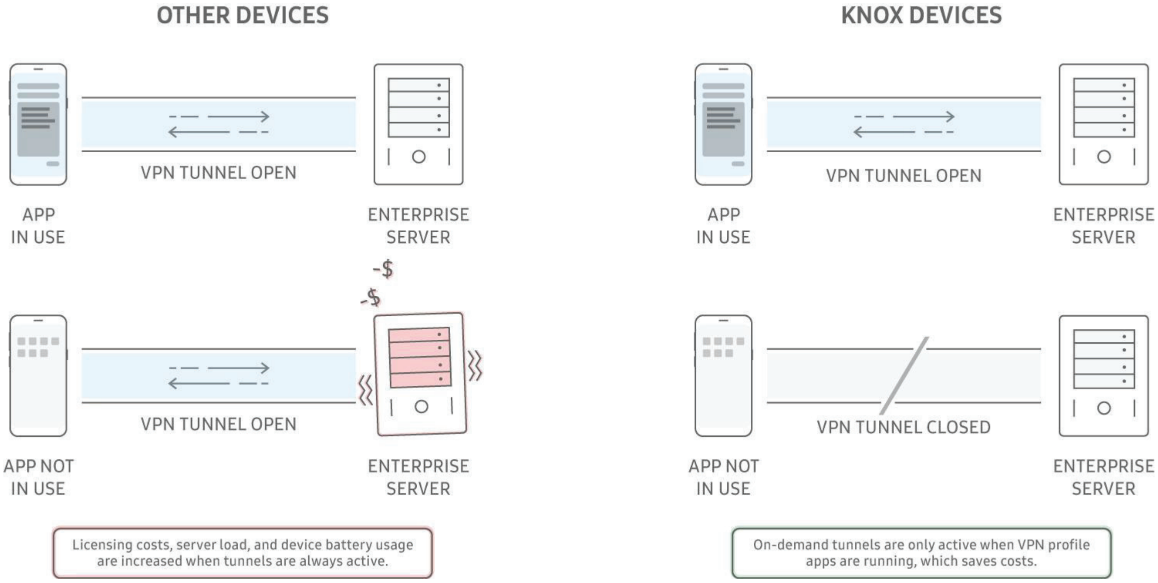
Virtual Private Networks (VPN)

Standard Android comes with basic VPN abilities that are adequate for most consumers. But many enterprises need better security and more flexible VPN controls for larger deployments. The Knox VPN framework includes the most advanced enterprise-focused feature set, which ensures that VPN connections are efficient, reliable, secure, and compliant with industry regulations and best practices. The Knox Platform VPN framework allows the integration of third-party VPN clients in addition to the built-in VPN client.

Unique advantages of Knox VPN framework

The Knox Platform VPN framework supports all common VPN types, protocols, and configuration options. When deploying VPN solutions, enterprise IT admins must ensure VPN deployments work smoothly, don't waste server resources, limit the VPN solution licensing costs, and enforce strict security policies that prevent data leakage.

The following is an example showing how Knox on-demand VPNs save cost:



The Knox Platform provides the following differentiating VPN features and advantages:

- The flexibility to use a VPN tunnel for the entire device, the Knox Workspace only, or a single app only.
- The unique ability to use a single VPN tunnel for traffic both inside and outside the Knox Workspace, without requiring separate VPN clients and licenses.
- The cost saving benefit of using VPN tunnels on-demand, only when apps in a VPN profile are running.
- The convenience to bypass VPN tunnels when a device is on-premise in a local corporate network.

- The strict coverage of corner cases to prevent data leakage outside of VPN tunnels, even during a device boot.
- The ability to connect multiple tunnels simultaneously.
- The extra security of chaining VPNs (also known as cascading or nesting VPNs) for greater anonymity, for example, in classified deployments.
- The power of configuring web proxies over VPN:
  - Web proxy configurations are tunnel-specific.
  - Web proxy support for NTLM authentication, basic authentication, PAC, and PAC with authentication.

The following Knox VPN features are also available, but are dependent on the VPN client:

- **QoS or traffic tracking and shaping.** The Knox VPN framework can inform the VPN client when any installed apps generate any traffic.
- **Automatic reconnection of VPN tunnels when the server side disconnects.** Server-side disconnections are more difficult to detect and handle than device-side disconnections, which are usually related to detectable conditions like loss of connectivity or the presence of new network connections, such as a new Wi-Fi connection.

Robust handling of enterprise requirements

Regardless of the features you choose, the VPN should act predictably even when the unexpected occurs. The following are some common scenarios where Knox Platform enhancements ensure proper VPN behavior:

- During a download, VPN tunnels direct download manager traffic to the VPN tunnel tied to the app that requested the download.
- VPN tunnels handle system events such as power saving mode entry or exit, package addition or removal, connectivity changes, and admin app changes.
- VPN profiles can specify which non-present apps must also use a VPN tunnel if they are ever installed.
- Even the free, built-in VPN client supports all the advanced VPN features listed in the previous list items.
- Robust blocking rules prevent data from leaking to the outside of the tunnel. Common gaps in coverage that Knox Platform VPNs correctly handle include:
  - A VPN client crash or other client app issues
  - A tunnel that has not yet been established, for example, during boot
  - A VPN client that is unable to connect to a VPN server
  - A proxy port that is blocking
- Handle captive portal prior to VPN tunnel establishment.

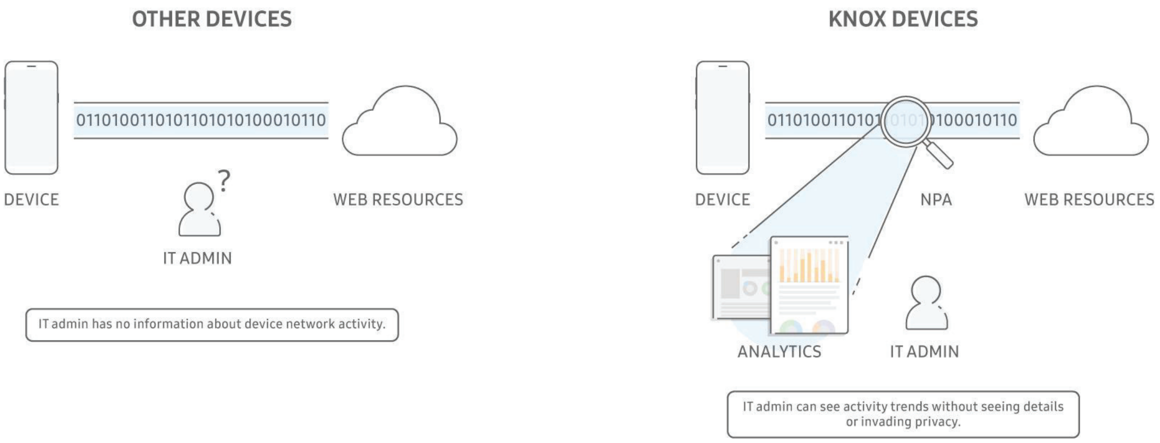
High-security built-in VPN client

The built-in Android VPN client (also called StrongSwan) is available on all Samsung devices, and is also integrated within the Knox Platform VPN framework, enabling the extra properties available within the Knox platform. The built-in VPN client, even without the Knox VPN framework, is differentiated from what Android offers, providing these advanced VPN features:

- FIPS 140-2 certified device cryptography components
- CPA certification at the Foundation grade, based on its successful Common Criteria evaluation against the Protection Profile for IPsec VPN Clients v1.4
- Security characteristics of IPSec VPN client version 2.5, as set by the NCSC
- Internet Key Exchange (IKE and IKEv2) and Suite-B algorithms:
    - IPsec IETF RFCs – IKEv1
    - IKEv1 – Main and aggressive IKE exchange modes with pre-shared key, certificates, Hybrid RSA, and EAP-MD5 authentications
    - IKEv2 with PSK and certificate-based authentication
    - IKEv2 – Pre-shared key, certificates, EAP-MD5 EAP-MSCHAPv2 authentication methods, and mobile extensions
    - Triple DES (56/168-bit), AES (128/256-bit) with MD5 or SHA
    - IKEv1 Suite B Cryptography supported with PSK and ECDS signature-based authentications
    - IKEv2 Suite B Cryptography supported with ECDSA signatures

Network Platform Analytics (NPA)

Endpoint devices, such as mobile devices, are hard to monitor for security issues. Third-party apps can't inspect OS behaviors and networking patterns, something that is possible on desktop platforms. These limitations, combined with the prevalent use of endpoint encryption create an information "black hole". This information black hole makes it more difficult to detect misconfigurations, troubling network usage patterns, the misuse of enterprise resources, or other signs of issues that impact an enterprise's bottom line.



IT admin has no information about device network activity.

IT admin can see activity trends without seeing details or invading privacy.

The NPA framework enables insights into mobile software and network use, misconfiguration, and network-based threats. Powerful analytics solutions use the NPA framework to increase endpoint visibility without violating the confidentiality of data moving across enterprise devices and networks.

Combined with a compatible analytics solution, NPA simplifies many device administration tasks:

- Detect **more** IT problems — "I don't know what I can't see!"
- Detect problems **faster** — "Notify me automatically of suspicious patterns."
- Investigate more **easily** — "Walk me through the chain of events."
- See root cause **attribution** — "Am I being attacked? Is this a bug? Is something misconfigured?"

- Provide **visibility** required to **trust** mobile devices — "Show me how my network is being used."
- Enable quicker **remediation** — "Lock down the device, user, or app causing this issue!"

NPA design

The NPA framework provides real-time information about the network packets leaving a device and the context surrounding the flow of data. An NPA-compatible Network Analyzer then analyzes the available data to provide valuable insights. Is your new beta app sending sensitive data to an unexpected server in a foreign country? Analyzing endpoint flow data gives us insights into network traffic, such as:

- The destination of every network flow, using either IPv4 or IPv6 addresses
- The domain name originator associated with the destination IP address
- The start and stop time for the network flow
- The number of bytes transferred in and out during the network session
- The name of the process or app initiating the data flow
- The cryptographic signature of the app initiating the data flow, and of its parent process
- Whether or not traffic originated from a tethered device (for example, a mobile hotspot) or from within the device

NPA maintains enterprise data confidentiality as it only inspects the header data and the context surrounding network traffic patterns. NPA and NPA-compatible network analyzers don't have access to actual data packets. This feature is a strong differentiator compared to solutions that unnecessarily collect and redirect all endpoint network traffic, usually by means of a web proxy or VPN.

Unique advantages of Knox NPA

The Knox platform NPA provides the only mobile platform for granular endpoint networking insights. Some unique advantages are:

- NPA is unaffected by endpoint network encryption.
- NPA can uniquely attribute network patterns to the specific software responsible.
- NPA can differentiate between traffic originating from a well-known Android app and a fake app impersonating the app.
- NPA does not expose your entire network traffic to the analytics solution.

NPA-compatible solutions

Samsung's release partner for NPA is Cisco. Cisco's network security products can now interface with Knox NPA to provide endpoint visibility of Knox devices. Admins can get this visibility even when a VPN is encrypting endpoint traffic. These insights are exposed to admins using the Cisco StealthWatch console and remediation steps performed using Cisco ICE.

Other Knox partners are preparing NPA-based solutions to help solve other common problems associated with mobile device deployments.

Certificate Management

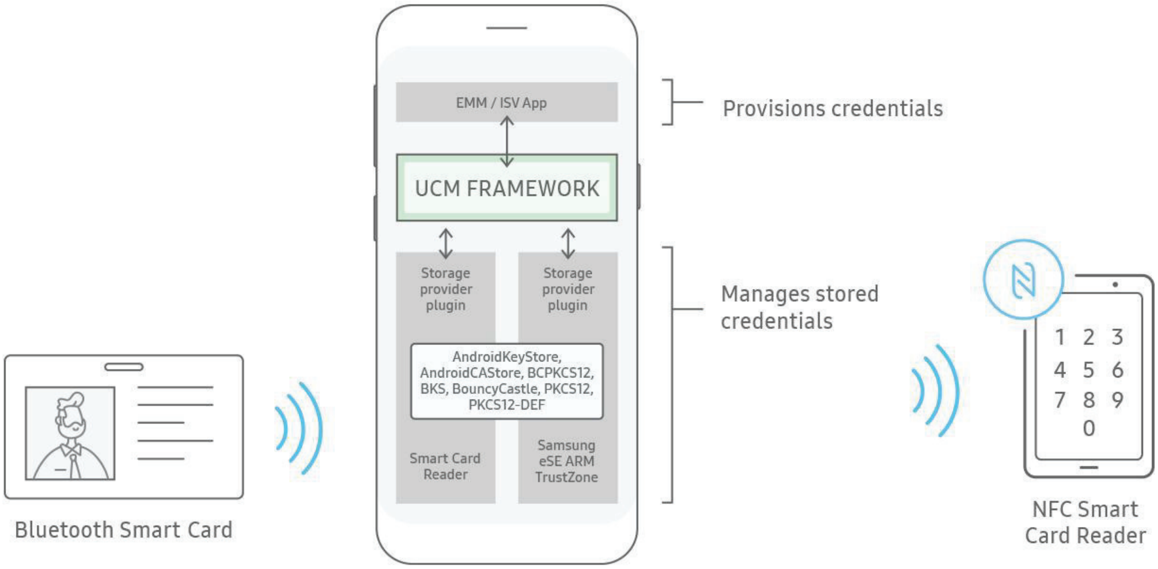
Universal Credential Management (UCM)

Digital credentials are critical mobile security building blocks, leveraging trusted authorities to validate identity and secure private channels across public deployments. Your mobile device credentials provide seamless access to secured Wi-Fi, VPN, email, and websites. Credentials include certificates providing identity and private keys to decrypt sensitive data. These credentials must be securely stored to prevent malicious parties from exploiting your identity and accessing confidential data.

The storage available to you can evolve with the introduction of new technology, and emerging security standards. For example, a mobile device used in a regulated industry may need to obtain personal credentials from a physical Smart Card. In the future, it may need to switch from physical smart cards to virtual ones on an NFC chip. This change process presents a fragmentation problem for credential consuming app developers, since each storage provider has its own proprietary APIs, so adding or switching to new storage hardware introduces new coding cycles, testing, and app re-distribution.

UCM framework

The Knox Platform's Universal Credential Management (UCM) provides a plug-and-play framework for credential management across a variety of different storage media. A significant benefit of the UCM framework is that it uniquely enables storage vendors to develop a plugin, distributed as a standard Android app, that provides access to their storage space and cryptographic operations without forcing app developers to change their code or forcing IT admins or end users to update their apps. The plugin essentially acts as the link between the UCM framework and a specific storage device.



The UCM framework consolidates and standardizes credential services to provide a streamlined interface for:

- **EMM or ISV apps** — These apps configure, provision, and consume credentials, managing credential storage access permissions, and activating advanced UCM permissions. The apps can enforce the installation, removal, or per-app access control of a credential.
- **Storage provider plugin** — These apps are provided by storage vendors to link the UCM framework to their storage solution, to manage stored credentials.
- **Secure storage** — This feature currently includes the Samsung eSE and Smart Card readers described in **"Secure storage options"** on page **30.** You **can easily support other storage options through additional vendor plugins.**

The Knox SDK provides credential storage vendors a set of UCM APIs to make current and future storage options available on Samsung devices, hiding the implementation details of their solution so that mobile app developers can transparently access stored credentials through standard APIs, such as the Android Keychain. Similarly, developers can use the Java Cryptography Extension APIs to offload cryptographic operations to a capable Smart Card. This abstraction, made possible by the UCM framework, eliminates the need for complex vendor-specific code within mobile apps, meaning enterprise customers have a wide range of existing apps available to them and can easily develop in-house apps without worrying about the underlying storage implementation.

Secure storage options

The UCM framework supports the following secure storage options:

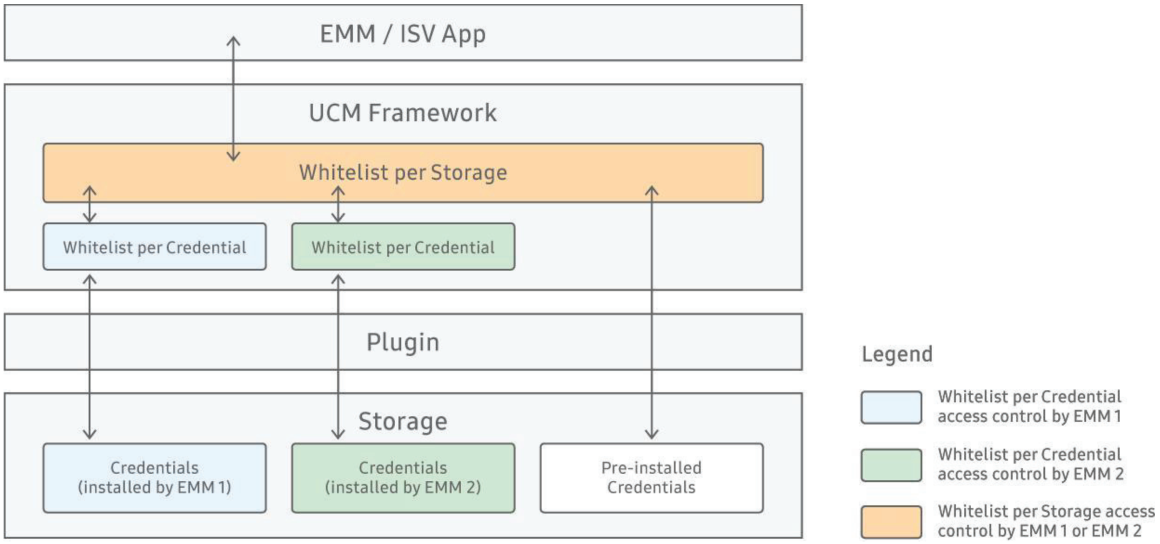
- **Samsung Embedded Secure Element** (eSE) — eSE supports the storing and accessing of credentials, allowing secure storage on the device without additional hardware.
  Note — eSE is not available with the following countries and carriers: USA-Verizon, Korea-All, Japan-All, Canada-Telus.

- **Smart cards** — Smart cards' resiliency makes them ideal for storing credentials if the threat model calls for trust to be shifted outside the device. You can use Smart Cards for unlock actions such as:
  - **Knox Platform's On Device Encryption (ODE)** — You can configure ODE to depend cryptographically on the PIN unlock of a Smart Card inserted in the device, which manages the decryption key for the internal data partition.
  - **Device lockscreen** — You can store the device unlock passcode in a Smart Card.

UCM whitelists

The UCM framework uses two types of whitelists, which uniquely manage access controls for credential storage and offer fully customizable access permissions:

- **App whitelist** — Enforces which apps can access each secure storage type. Every secure storage device maps to its respective UCM plugin, that a secure storage solution provider creates and maintains.
- **Credential whitelist** — Enforces each app's access to credentials, providing app-specific access permissions. By enforcing access control, admins can prevent credential usage by malicious or untrusted apps.


EMM / ISV App
UCM Framework
Whitelist per Storage
Whitelist per Credential
Whitelist per Credential
Plugin
Storage
Credentials (installed by EMM 1)
Credentials (installed by EMM 2)
Pre-installed Credentials
Legend
Whitelist per Credential access control by EMM 1
Whitelist per Credential access control by EMM 2
Whitelist per Storage access control by EMM 1 or EMM 2

Client Certificate Manager (CCM)

Samsung builds upon the Android Keystore by providing a tamper-proof, detection-based lock-down of cryptographic keys and certificates. This solution supports a variety of high-security use cases important to enterprises, as described in the following sections.

Granular certificate and key access control

The Knox Platform supports an app whitelist for certificates, allowing the certificate installer to define which apps are allowed to perform cryptographic operations based on their certificates. This certificate whitelist process offers better control and flexibility than simply allowing app-only or device-wide access rights to certificates.

Signing with device-specific certificates

A special certificate called the **Device Default Certificate** (DDC) resides within each device. What makes this certificate special is that it is tied to that device's hardware, is signed by the Device Root Key (DRK), and can never leave the device.

Any objects signed by the same DDC are guaranteed to have come from the same Samsung device. There is no way to spoof the identity of a device by reusing a DDC and its key pair on a different device.

Device integrity assurance

Objects signed with this certificate were signed while the device was in good health, meaning when the device was uncompromised. If a device fails its integrity checks—by failing the signature check of the kernel or OS or disabling SE for Android—the following happens:

- A tamper fuse is set; and
- The DDC is rendered permanently unusable.

This lockdown helps attest to the health of the device where the data was signed. After all, you can't trust a signature if the device doing the signing is compromised. The Knox Platform provides a CSR agent that benefits from this device health attestation claim. A CSR produced and signed by the CSR agent carries implicit device health security claims.

Keystore integration with other features

A keystore is only as useful as the use cases it supports. In addition to manual cryptographic actions— such as sign, verify, encrypt, and decrypt—the Knox Platform provides built-in logic to support sensitive certificate-based actions enterprises often need to secure their solutions such as the following:

- **Certificate Signing Requests** (CSRs) — The ability to complete CSRs with a trusted agent, tied to the Knox Platform's hardware-based Root of Trust, simplifies the secure handling of mobile endpoint requests for digital identity certificates. Instead of sending key pairs and certificates from servers, keys can instead be securely generated on-device and bound to hardware. The public certificate is then included in an appropriate CSR request. Using the CSR agent to validate CSR contents and sign the request avoids trusting sensitive actions to third-party code running in less trusted areas of the device.
- **Certificate Enrollment Protocols** (CEPs) — Similar to CSR, CEP provides built-in agents for logic that enterprises rely on, saving time and enhancing security claims. For more information, see Certificate Enrollment Protocols.

In addition to the DDC, you can generate or install your own certificate and key pairs and specify they are accessible only if the device is in good health. This additional process locks down the keystore in the event of a device integrity failure.

Certificate Enrollment Protocols (CEP)

The Certificate Enrollment Protocols (CEP) provision and support digital certificates for apps within Samsung devices. This feature is of great assistance to EMMs and third-party vendors. Why? Because the CEP helps complete certificate enrollment without device user intervention, further solidifying the claim that Samsung Knox devices provide both world-class security as well as industry-leading manageability.

Enterprises can use CEP to:

- Enroll, renew, or delete certificates
- Check your deployment's certificate enrollment or renewal status

The CEP service is very robust, and supports the following enrollment protocols and standards:

- Simple Certificate Enrollment Protocol (SCEP)
- Certificate Management Protocol (CMP)
- Certificate Management over Cryptographic Message Syntax, Enrollment Over Secure Transport (CMC-EST)

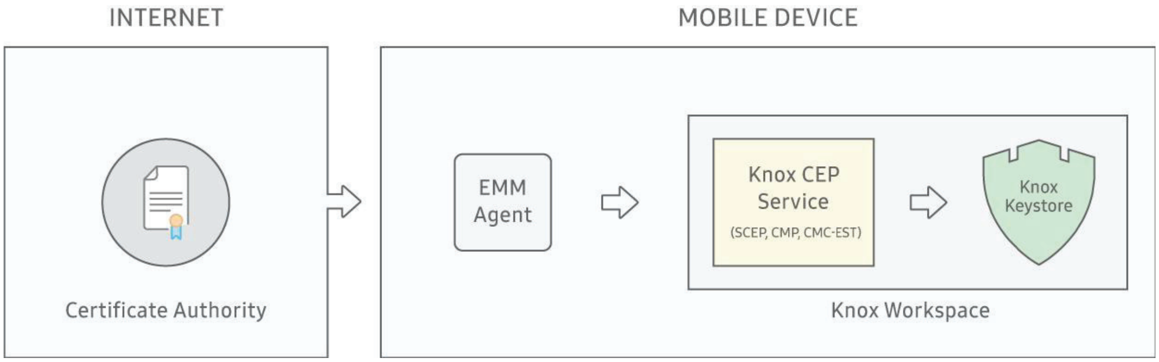
SCEP, CMP, and CMC are frequently used certificate enrollment protocols for provisioning digital certificates. For more information on these protocols, see Internet Engineering Task Force (IETF).

CEP asymmetric key acquisition

Apps use CEP to acquire the public part of an asymmetric key. Asymmetric keys have a public part and a private part. The private part never leaves the Keystore, but the public part is freely distributed. The key owner can use the Keystore to apply the private part of the asymmetric key to an encrypted message to decrypt it.

CEP operational environment

CEP functions within the scope of either the Knox Workspace or personal space, depending on where it is installed. If the deployment objective is to provision and manage certificates for apps inside the Knox Workspace only, then you must install the CEP services within the Knox Workspace as follows:



If the objective is to provision and manage certificates for apps in the personal space, then you can install the CEP services in the personal space to provision and manage certificates.

EMM agents can call the CEP services in either the personal space or Knox Workspace. EMM agents don't have access to a service created outside their scope.

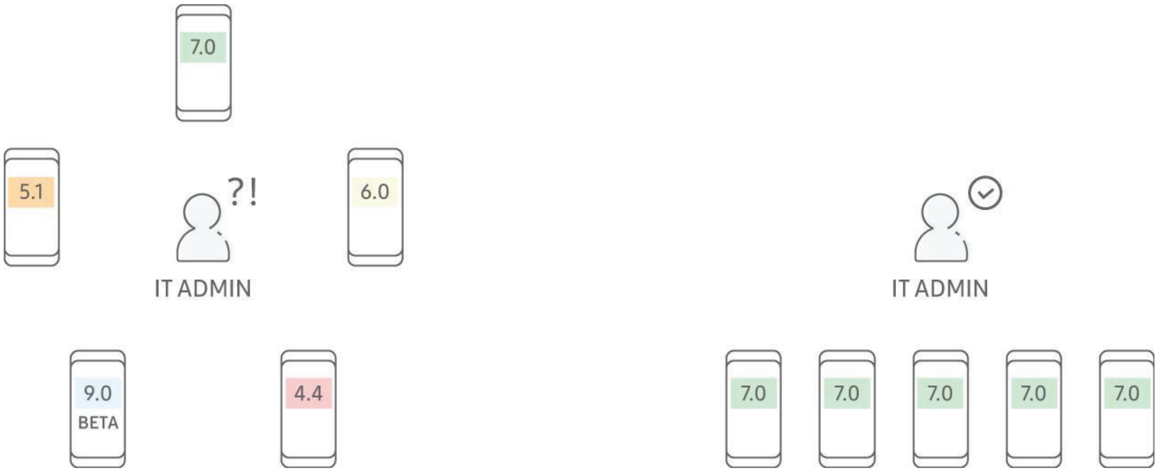
Device Management

Device Software Update Management

Frequent software updates are often necessary to resolve bugs, patch security vulnerabilities, and enhance device capabilities. But IT admins must understand and validate software changes prior to mass deployment. Samsung released the mobile industry's first firmware update management system on Android to enable IT to test and validate software updates and to control roll-out scope and timing.

Why manage device software updates?

In enterprises with fragmented platforms and firmware versions, mobile device deployment and support becomes a time-consuming and tedious task. Proprietary enterprise apps and websites behave inconsistently on different firmware versions, so features require testing and troubleshooting on a widening array of device platforms.



Controlling the rollout of software updates allows IT admins to:

- Homogenize the firmware versions and capabilities of deployed device models.
- Carry out interoperability or compatibility testing with in-house or proprietary servers, apps, and endpoint settings.
- Ensure that known issues are patched before deployment of major firmware version updates.
- Perform field tests of new firmware and software on a subset of devices before mass deployment.
- Force the use of firmware versions that have been validated to meet industry certification or regulation requirements.

Strict control over device firmware updates

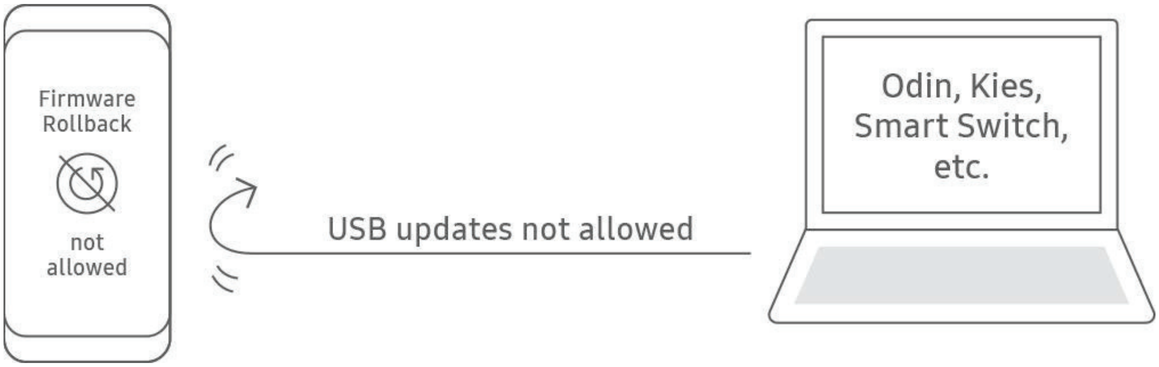
Samsung developed Enterprise Firmware Over-the-Air (E-FOTA) to enable enterprises to save time and support costs, and manage the mobile infrastructure as efficiently as possible.

With E-FOTA, enterprises can control device software updates as follows:

- **Select the highest firmware version allowed on devices** — This option ensures that device users can't independently update to an unsupported firmware version, preventing issues that could negatively impact employee productivity, support costs, and data security.
- **Force the download of a specific firmware version onto select devices** — Enterprises can download new firmware to a few test devices to run interoperability or compatibility tests. This mandatory download is done with proprietary systems and apps to find any corner cases that might result in operational or performance issues.
- **Mass deploy a new firmware version** — Mass deployment prevents software version fragmentation so IT teams don't need to support multiple legacy firmware versions for each deployed device model.
- **Schedule updates during non-peak work times** — This option ensures updates don't interfere with employee productivity.

Knox control over user updates

A wide range of EMM partners support Samsung's firmware management features, integrating firmware management with other asset management activities. IT admins can use these tools to test and deploy software updates in a consistent and low-risk way. Through EMM solutions, enterprises can restrict users from loading unauthorized firmware, through their devices or USB-connected computers.



Through the Knox Platform, enterprises can:

- **Prevent firmware rollback** – This option prevents valid, but out-of-date firmware versions from being maliciously or accidentally installed onto an enterprise's devices. On Samsung Knox devices, a Rollback Prevention fuse encodes the minimum acceptable version of Samsung-approved software. With specific updates, the next set of fuses are burned to indicate the new update is now the minimum version allowed to boot. You can't disable this basic, built-in security feature.
- **Disable automatic firmware updates** – IT admins can prevent users from going to their Android Settings to enable or disable automatic firmware updates.
- **Disable all OTA updates** – IT admins can prevent users from going to their Android Settings to enable or disable software updates in general. This restriction includes updates for firmware, security patches, bug fixes, and apps.
- **Disable USB-connected updates** – IT admins can prevent users from booting into Download Mode and installing a manual software update. This restriction includes updates through the Odin, Kies, and Smart Switch update tools.

Granular Device Management

The Knox Platform's granular device management features are specifically curated, from partner feedback and industry data, to solve some of the most common frustrations enterprises face when mass deploying devices. These unique policies provide device flexibility and customization beyond any other device provider. The policies help organizations manage operations more effectively, secure confidential assets, and reduce administrative overhead. They also solve particular issues regarding industry regulation and compliance. For example, Rich Communication Services (RCS) logging is required by law in the financial industry. Samsung is the only vendor to provide this critical auditing feature.

Custom boot banner

Samsung Knox is the only mobile platform that allows an enterprise to natively change the device boot logo. In many industries, such as government or defense, this change is mandatory for compliance. Through the Knox Platform, enterprise IT admins and developers can customize the following:

- Samsung boot up display
- Splash screen animation, when the device is turned on or off
- Lockscreen image, which can provide an enterprise logo or contact info for lost phones

Enterprises can use these capabilities to mitigate problems such as the following:

- **Phone is lost and found** — Owner information is available by simply powering on the device. There is no need to attempt to unlock the device or call the carrier. The device can be returned to the enterprise quickly.
- **Multiple phones** — Displaying an enterprise logo on bootup lets users know that the device belongs to and is secured by the enterprise. This logo clearly distinguishes it from other devices in the user's possession.

Split billing (Dual APN)

Split billing separates enterprise and personal data usage.

- In Bring Your Own Device (BYOD) deployments, enterprise billing allows employees to be properly compensated for data costs generated from work-related app usage.
- In Corporately Owned, Personally Enabled (COPE) deployments, enterprise billing allows employers to pay for data usage incurred only for work purposes.

Split billing also works with dual SIM devices, by mapping some apps to using the data plan from one SIM, and other apps to the other SIM's data plan.

Remote admin lock of device

This feature allows an IT admin to remotely lock out a device, for example, when the device is out of compliance. Once the device is locked, only an IT admin can unlock it and not a device user. This functionality solves two problems:

- Prevents unauthorized users from accessing the device if it gets lost or stolen.

- Prevents users with valid login credentials from using the device, for example, if the credentials are stolen or the user is no longer allowed to use the device.

With stock Android, an IT admin can lock a device only if it is currently unlocked. If the device is already locked, an admin can't lock it to prevent future unauthorized logins.

Enterprise roaming

Roaming mobile connections can incur unexpected data costs. Multiplied across an enterprise's mobile workforce, these costs can become exorbitant.

Rather than just simply disabling all mobile roaming, the Knox Platform provides more granular controls for enterprises, letting them control which mission-critical apps are allowed to use data during mobile roaming. Enterprises could enable roaming data for:

- All apps in the Knox Workspace
- A single app within the Knox Workspace
- A single app in the personal space

They can also set up Split Billing, with separate roaming policies for the APNs set up for personal and enterprise billing.

Granular policies

Call restrictions

Enterprises can apply granular settings to the caller app, allowing only:

- Emergency calling
- Calling to certain numbers
- A limited number of calls per day

RCS logging

The Knox Platform allows an enterprise to log RCS messages. For many industries, such as financial services, the ability to audit sent and received messages is required by law.

RCS messaging is a new messaging protocol that replaces SMS as the default messaging platform for carriers. It adds much needed features such as group messages and allows users to send more file types. Currently, enterprises that can't capture RCS messages must turn RCS off and lose the benefits of this new protocol. Knox RCS logging capabilities mean deployments can use powerful RCS abilities while staying compliant.

SMS management

Knox provides many advanced SMS policies. Policies frequently used by organizations include:

- Adding an automatic company disclaimer to the bottom of every outgoing text
- Restricting the number of texts per day

- Auditing and recording all incoming and outgoing SMS messages

SD card restrictions

Most vendors don't provide sophisticated options to manage an SD card. Typically, enterprises must choose between one of two options: allow full read and write access to the SD card or completely block it.

The Knox Platform addresses this industry pain point by giving enterprises independent control over read and write access. Knox can:

- Allow read access but block write access
- Allow write access but block read access

This level of control means you can provide one-way data access to sensitive data to effectively meet your security requirements.

Bluetooth restrictions

To mitigate attacks perpetrated through Bluetooth connections, Knox provides these controls:

- **Completely disable Bluetooth** — Turn off Bluetooth and Bluetooth background scanning.
- **Block specific Bluetooth profile types** — Restrict the types of Bluetooth devices that the user can connect to the device, for example:
  - Allow Bluetooth headphones
  - Block Bluetooth file transfers, which could leak private data

USB class restrictions

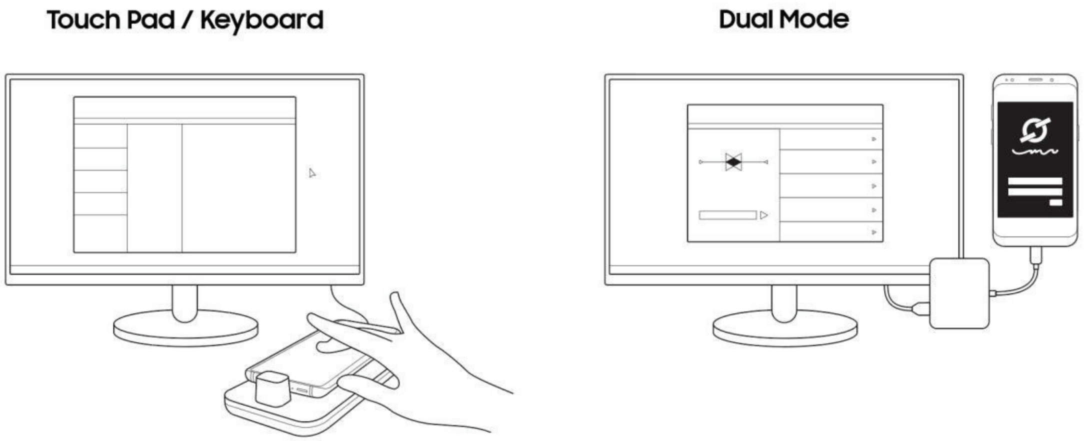
Knox can restrict or allow different types of USB-connected devices, more specifically, the USB device classes defined through usb.org. This feature includes access to the following USB device classes:

- Audio, Video, Audio/Video
- Mass Storage
- Content Security
- Smart Card
- Printer
- Hub, Type-C Bridge, Wireless Controller
- Human Interface Device (HID)
- Communications, CDC Control, CDC Data
- Personal Healthcare
- Billboard
- Diagnostic

For example, you could block all USB devices except Smart Card readers.

Samsung DeX Management

Samsung DeX is a unique product that lets you use your phone as if it were a laptop or desktop computer. You simply connect your phone to a monitor, and optionally use a mouse, keyboard, or S-Pen to launch apps, move objects, type text, write text, or draw images.



DeX supports two different modes:

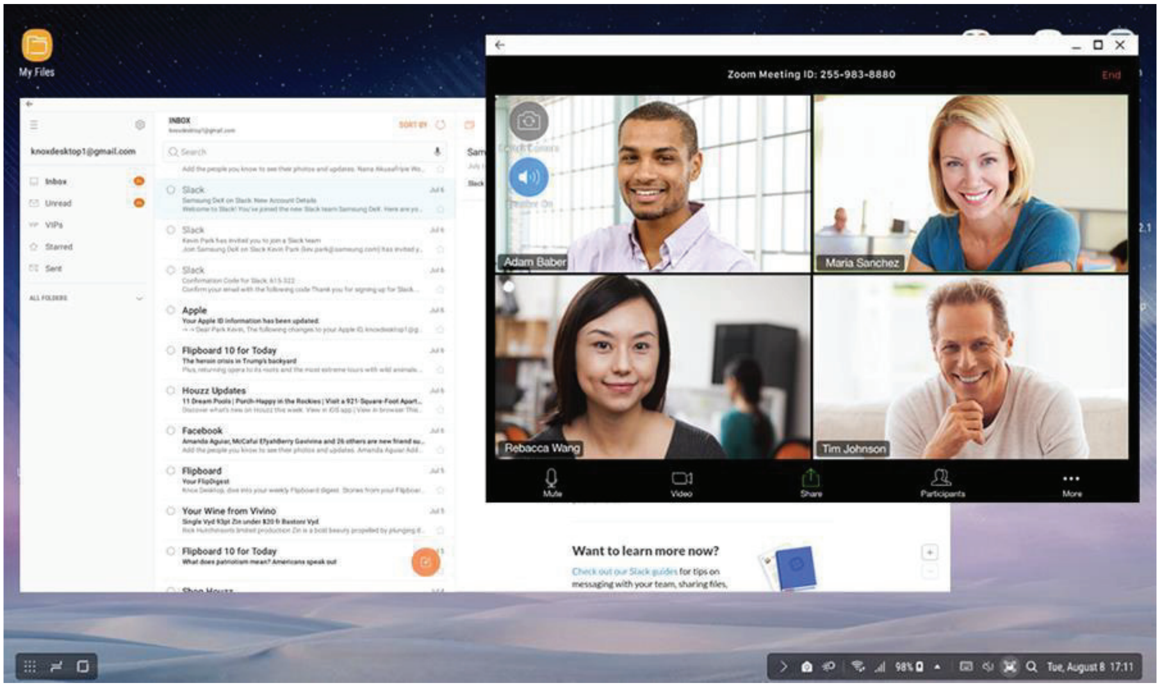
- **Touch Pad/Keyboard** — The phone's screen appears on the connected monitor. You can use the phone's touch pad or a connected keyboard and mouse to enter text and move the cursor on the monitor. Use this mode to read or write documents, participate in video conferences, compose more complex emails, edit images, or develop slide presentations.
- **Dual Mode** — You can use both the phone screen and the monitor at the same time.

You can connect your phone to a monitor and peripherals using one of the following options:

- Samsung DeX docking station
- Samsung DeX docking pad
- USB-C to HDMI adapter

Why use Samsung DeX?

Instead of having to carry both a laptop and phone, you now need only a phone. Through a single portable device, you can quickly write documents, edit spreadsheets, and create presentations on a conventional large screen. There is no need to purchase or carry along a separate laptop. The DeX mode untethers employees from their laptops, and offers enterprises many capital cost savings opportunities.



Using Knox to customize DeX

Enterprises can use the Samsung Knox platform to secure the way Samsung DeX works, allowing them to benefit from the Knox Platform's defense-grade security features without sacrificing the innovation and productivity that comes with DeX.

Using a large screen in DeX mode means that sensitive information may be visible to passersby. As such, you can use the Knox platform to improve security in DeX mode. You can deploy security policies such as:

- Setting a screen timeout while in DeX mode
- Allowing only Ethernet connections, no Wi-Fi or cellular data
- Disabling specific apps in DeX mode, for example, apps displaying confidential data
- Disabling DeX mode

You can also use the Knox Platform to customize the DeX interface. Available customizations include:

- Uploading a company logo to the DeX loading screen
- Adding or removing shortcuts from the DeX launcher

Unique advantages of Samsung DeX

- **Mobile desktop experience** — Enables phone use, on the go, in a desktop environment. A separate laptop is unnecessary. You can access the apps and files necessary directly using your phone.
- **Defense-grade security on a desktop** — Protects users and enterprises with industry-leading security while preserving the productivity enhancements of a desktop environment.

- **Universal app compatibility** — Compatible with the native Samsung and Android apps that are pre-installed on devices. Popular apps such as Microsoft Office apps and Adobe Photoshop Express are also optimized for use with DeX to take advantage of larger, multi-app displays.
- **Customizable** — Mobile app developers can enhance and control their apps while in use with DeX, using DeX APIs from the Knox SDK.

Firewall Management

Most mobile device platforms use built-in firewalls, but don't provide granular control over firewall settings and activity. With the Knox Platform, you can deploy firewall configurations specifically catered to your enterprise security needs.

Why manage and customize device firewalls?

Default firewalls may not provide your organization with the security and data protection it needs. In fact, some firewalls may not even let you see the rules they are enforcing. However, when configuring firewalls with the Samsung Knox Platform, you can know exactly what policies are deployed and take additional measures to secure your enterprise systems.

With the Samsung Knox Platform, you can:

- Restrict and redirect Internet access to specific IP addresses and domains
- Set firewall policies on a per-app or device basis
- Produce logs reporting the blacklisted domains that users accessed

Granular control of Internet access

You can limit the permissible network connections to only trusted addresses, by setting the appropriate Internet access restrictions. The Knox Platform offers a variety of restriction methods, all of which can be used together:

- **IP address filters** — Allow, deny, and redirect access to specific IP addresses. Configure a filter to apply to transmitted data, received data, or both. Allow or deny both IPv4 and IPv6 formatted addresses.
- **Domain name filters** — Allow or deny access to an entire domain or sub-domain.
- **Per-app and device-wide modes** — Give specific apps—for example, ones that handle confidential data—stronger firewalls, and all other apps on a device a more lenient firewall configuration.

Log unsafe URL access

The Knox Platform provides visibility into denied attempts to access blocked domains. The improved visibility helps you to remain aware of potential security breaches or insecure browsing practices within your organization.

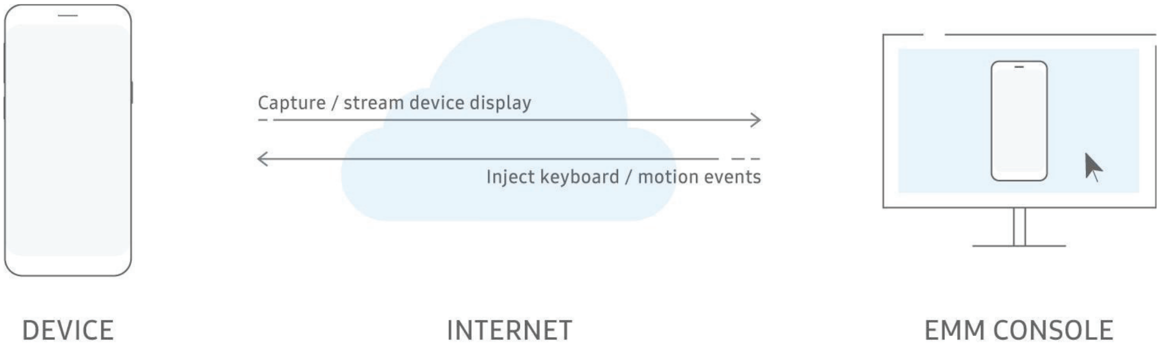
The Knox Platform logs reports with the following information:

- **App name** — The package name of the app attempting to access a blocked domain.

- **Blocked domain URL** — The URL of the domain name blacklisted by your firewall.
- **Timestamp** — The time the incident occurred, to assist with troubleshooting incidents.

Remote Control

With the increasing complexity of problems that IT admins must solve, Knox Remote Control provides IT admins a powerful way to quickly and remotely fix issues. Not only can IT admins have real-time access to what the remote device screen is displaying, but also control it by injecting actions such as finger, keyboard, and mouse events. Although other mobile platforms also offer remote viewing of remote device displays, only Knox provides built-in remote control of devices without requiring third-party solutions.


Capture / stream device display
Inject keyboard / motion events
DEVICE
INTERNET
EMM CONSOLE

Here is an example use case: An enterprise employee is on a business trip. On encountering a problem with the company-issued mobile phone, the employee contacts an enterprise IT admin. The IT admin uses an EMM console to remotely view the device screen to observe the issue first hand, then remotely controls the device, through finger, mouse, or keyboard actions. The IT admin directly accesses the environment to remotely debug the issue on the device. The employee is now quickly productive, without the frustrating downtime associated with relaying instructions verbally or through email.

The continuous polling of the device screens doesn't impact device performance as devices send only screen changes.

Unique advantages of Knox Remote Control

The Knox Platform provides built-in remote control without requiring third-party solutions. For enterprises, this control:

- **Saves time** by enabling IT admins to troubleshoot remote mobile device issues in real-time and utilize high performance screen sharing.
- **Reduces employee down-time** and **optimizes employee productivity** through quick problem resolution.
- **Enables monitoring** of devices for corporate policy violations along with corrective actions on the devices, as well as an ability to monitor only for screen changes.

Audit Log

Organizations that need to troubleshoot serious security breaches rely on audit logs for a forensic analysis of the activities leading up to actual and potential breaches. In regulated industries, these audit trails are a mandated requirement to comply with security audits.

With the Knox platform, an enterprise IT admin can use an EMM console to enable audit logging on all corporate devices. IT admins can proactively pull audit logs from time to time, to detect and defend against malware or viruses at the earliest onset. In the event of a possible intrusion, IT admins can parse the logged events for unauthorized activities.

The Knox platform Audit Log provides comprehensive information about device events, including:

- Knox Workspace container activities
- Password policies set for devices and containers
- App installation and removal
- Certificate failure and key generation
- Account creation and removal
- File exchange attempts over Wi-Fi

To help better manage device storage, IT admins can control the Audit Log size.

The benefits to an enterprise include:

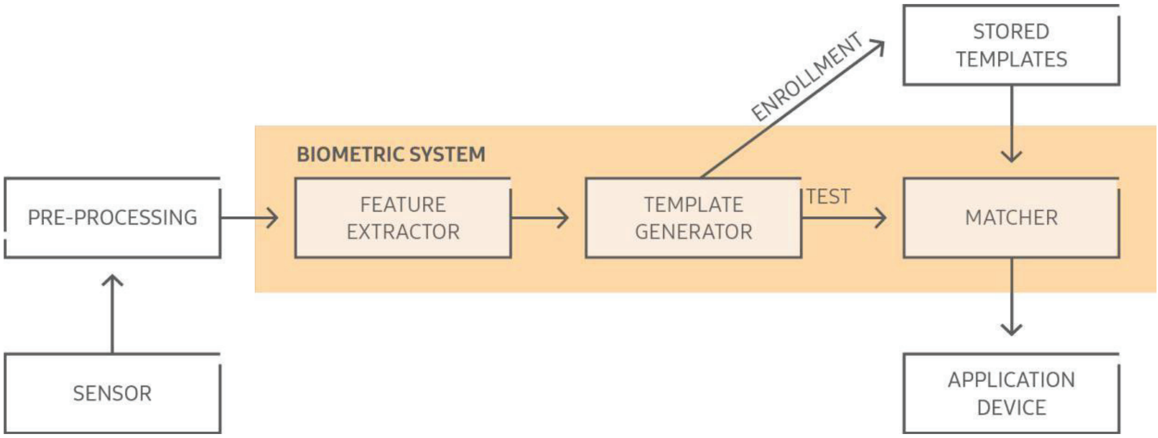
- Early detection and defense against malware and viruses.
- Empowering IT admins with powerful troubleshooting data.
- Adherence to mandated requirements in regulated industries.
- Compliance with the Mobile Device Fundamentals Protection Profile (MDFPP) 2.0 requirements to collect events.

User Authentication

Biometric authentication

Traditional user authentication relies on things you know or have, like a password or ID card. These are susceptible to human mistakes, phishing, and duplication. Biometric authentication validates a personal trait, for example: fingerprints, irises, or facial features. Biometrics can lower the false acceptance rate (FAR). Users can use biometrics to unlock devices and app containers. Through Samsung Pass, users can also use biometrics to log into apps and websites.

Advantages of Knox Biometrics



The Knox Platform provides the following in addition to standard Android capabilities:

- **Secure storage** — On Samsung devices, the authentication software doesn't share or distribute the biometric measurements of any user. The measurements are stored in a format that can't be used to reproduce the original biometric, and can only be accessed and decoded within the specific part of the TrustZone that has access to the biometric hardware. Biometrics are used only on the correct device and by the correct user. This functionality means there is a lower chance of someone spoofing biometrics credentials to access a device.
- **Enforced two-factor authentication** (2FA) — The Knox Platform provides IT admins the option to enforce two-factor authentication with biometrics for the Knox Workspace. For example, a user can be required to authenticate with an iris scan in addition to a standard device unlock method (password, PIN, pattern). While Android provides some combinations of two-factor authentication, the Knox Platform allows you to take your security one step further with biometric integration.
- **Samsung Pass integration** — Apps can use Samsung Pass APIs to enforce biometric authentication in place of a traditional login and password. This authentication method can save an organization a large amount of password management overhead, while further increasing device security. Samsung Pass features the ability to:
  - Support Fast IDentification Online (FIDO) authentication
  - Register and deregister a user’s biometrics
  - Respond to remote wipe requests
  - Manage authentication transactions

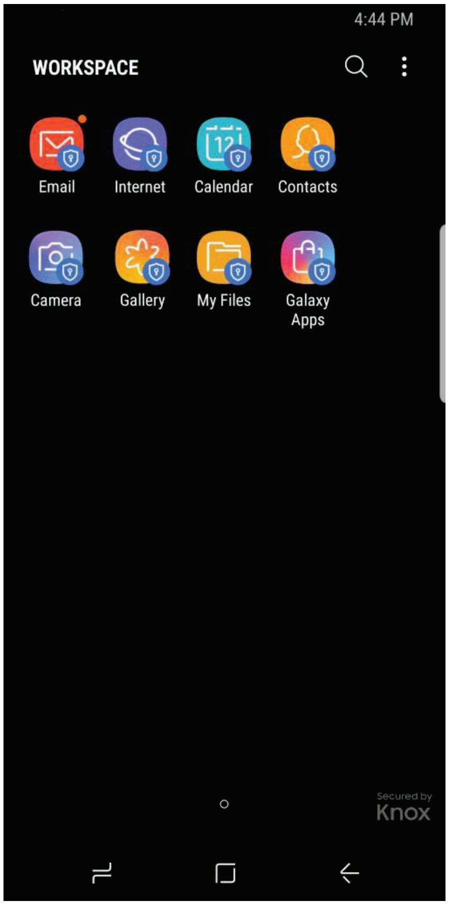
  - Work in the Secure World of the TrustZone
- **Enterprise credentials override** — As required by enterprise policy, Knox devices allow you to enforce the use of enterprise AD credentials to unlock a device or Knox Workspace container. This setting overrides any biometrics set by the user, and forces them to use their enterprise credentials.

App and Data Protection

Enterprise Productivity Apps

Mobile apps have changed the way we work by providing new channels of communication, innovating customer engagement, and empowering organizations with critical data in real-time. Samsung Knox devices include a set of productivity apps for both personal and business use.

Business-critical apps include Samsung Email, Internet browser, Calendar, and Contacts. Enterprise IT admins can secure these apps within the Knox Workspace, along with other apps used by the enterprise.



The Knox Platform secures enterprise apps and protects confidential app data through these methods:

- **App installations and updates** — Apps are pre-installed within the mobile device's secure Knox Workspace and users can update these apps independent of firmware updates through Google Play.
- **App isolation** — Apps are sandboxed within the Knox Workspace, which uses SE for Android to prevent personal apps from interfering with the business apps that are in the Knox Workspace.
- **App permissions** — Knox provides App Permission Monitoring to help users prevent malware from using powerful permissions to gain unauthorized access to the device and Knox Workspace.
- **Data-At-Rest** — Through Knox's Sensitive Data Protection (SDP), the files and data used by an app can remain encrypted until device users authenticate at device unlock or Knox Workspace login. Individual apps can further deploy an app-specific password as another line of defense.
- **Data-In-Transit** — App data sent through the public Internet can be secured using Knox's advanced VPN features.
- **DeX integration** — Not only are all Samsung native apps optimized to work within DeX, enterprises can secure apps while they're displayed in DeX.

Samsung Email

The Samsung Email app is uniquely designed for customers requiring the secure synchronization of their mobile device's Email calendar, tasks, and memo functions. The Email app can use MS Exchange ActiveSync (EAS) for Single Sign On using company credentials.

In contrast with third-party security solutions, the Samsung Email app uses Sensitive Data Protection (SDP) by default, to automatically:

- Protect email text and attachments
- Secure incoming emails and notifications in real time

The Samsung Email app provides these key benefits:

- Productivity
  - Single Sign On (SSO) with EAS
  - EAS synchronization of contacts, calendar, tasks, and note data
  - Federated LDAP query support
- Security
  - EAS certification for account
  - EAS certification for S/MIME messages
  - EAS certification revocation checks
  - EAS certification history support
  - Card certification support
- Management
  - LDAP account management
  - EAS account management

Samsung Internet Browser

The Samsung Internet Browser provides enterprises with the following security features:

- **Biometric Authentication** — IT admins can enforce biometric authentication for website logins, web payments, and accessing Secret Mode.
- **Secret Mode Password** — IT admins can enforce password access to Secret Mode, which can contain confidential bookmarks and saved pages.
- **Protected Browsing** — IT admins can enable warnings to alert users if they try to view known malicious sites, which might try to steal confidential data such as passwords or credit card information.
- **Content Blockers** — IT admins can allow the use of third-party plugins to filter out content such as:
  - ads, which can come with cookies, malware, or viruses
  - invisible trackers, which can monitor online activity

Enterprises can take advantage of the following additional capabilities to secure mobile browsing:

- Set up an HTTP proxy
- Enable TLS encryption of browser traffic
- Filter URLs or domains
- Block pop-ups through extensions
- Disable or enable JavaScript
- Disable or enable the auto-fill of forms
- Disable or enable cookies, saved sign-in data
- Delete or preserve personal data

Samsung Contacts

Contacts are the lifeline of any collaborative business environment and empower mobile workers to stay connected. Enterprises need to strike a fine balance between providing employees with easy access to contacts and protecting private contact information from exploitation.

The Samsung Contacts app provides enterprises with the ability to disable or enable the following features:

- Synchronization of contact data with an MS Exchange or ActiveSync server
- Synchronization of contact data inside and outside the Knox Workspace container
- Copying of contact info to a SIM card
- Accessing contact info at the end of a phone call

Advanced App Management

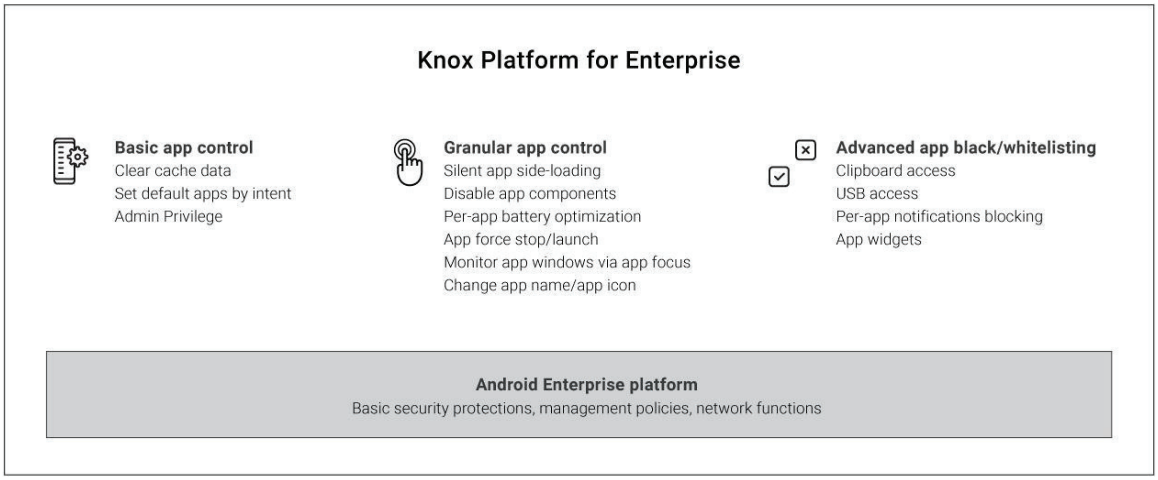
Enterprises need a strong Mobile Application Management (MAM) strategy to deploy apps effectively, manage app licenses, secure apps, optimize app usage, and handle app data safely. The Knox platform provides comprehensive app management capabilities that allow IT admins to control all aspects of apps installed on a device. These capabilities can also be extended inside the secure Knox Workspace to provide a safe haven for sensitive apps and data.

Enterprises use EMM solutions to centrally configure and remotely manage apps. Knox provides a full complement of management functions, providing IT admins with the ability to:

- Install, uninstall, update, enable, disable, start, stop, or wipe data for an app
- Whitelist or blacklist the following:
  - apps that can be installed
  - apps that can auto-update
  - apps that can use the Clipboard
  - apps that can be started and stopped by users
  - apps that can access the USB port
  - app accounts, permissions, and notifications
- Disable or enable other apps like Google Play, Google Chrome, Voice Dialer, and YouTube
- Get info like the app code size, cache size, data size, total size, notification mode, and restrictions
- Get statistics like app launch count, component state, app focus state, CPU usage, data size, memory usage, and network stats

Unique advantages of Knox App Management

What sets the Knox platform apart from other mobile platforms are the advanced app management features not found in other solutions, providing additional advantages that enable enterprises to be fully efficient and productive.



Knox Platform for Enterprise

Basic app control
Clear cache data
Set default apps by intent
Admin Privilege

Granular app control
Silent app side-loading
Disable app components
Per-app battery optimization
App force stop/launch
Monitor app windows via app focus
Change app name/app icon

Advanced app black/whitelisting
Clipboard access
USB access
Per-app notifications blocking
App widgets

Android Enterprise platform
Basic security protections, management policies, network functions

App control

- **Clear cache data** — Remove cache memory for an individual or list of apps to help optimize space and have complete control over your data.
- **Set default apps by intent** — Set an app as the primary app for a given task. For example, ensure your solution only uses a certain Internet browser or force your SMS service to comply with your strict company policies.
- **Admin privilege** — An admin can prevent the activation of another admin's app, unless the app is part of the whitelisted apps.

Advanced app black and whitelisting

- **Clipboard access** — Prevent access to the native Android clipboard within an app. If an app tries to use the clipboard, the content is deleted.
- **USB access** — Prevent user permission for one or more USB devices to be used by an app.
- **Per-app notifications blocking** — Prevent status bar notifications for an app and choose to block either text, sound, or both.
- **App widgets** — Allow only approved widget packages into your Knox Workspace container, and view them in launcher mode on a Samsung device.

Granular app control

- **Silent app side loading** — Silently install any app without user interaction or permission.
- **Disable app components** — Enable or disable a specific package component such as the activity, receiver, service, or provider class.
- **Battery optimization** — Whitelist apps from Google's Doze mode, app standby or power saving mode.
- **App force stop and launch** — Force stop any app including background processes and system apps.
- **App focus** — Monitor any app and receive a notification if a user leaves the window of a whitelisted app.
- **Change app name or app icon** — Change an app's package name and icon.

DualDAR Encryption

Protecting Data-At-Rest (DAR) on mobile devices is a major concern for security conscious enterprises. The Samsung Knox Sensitive Data Protection (SDP) already addresses this issue, by decrypting data only after user authentication, providing per-file and per-data decryption keys, offering per-app password checks, and meeting MDFPP requirements for US government and military use.

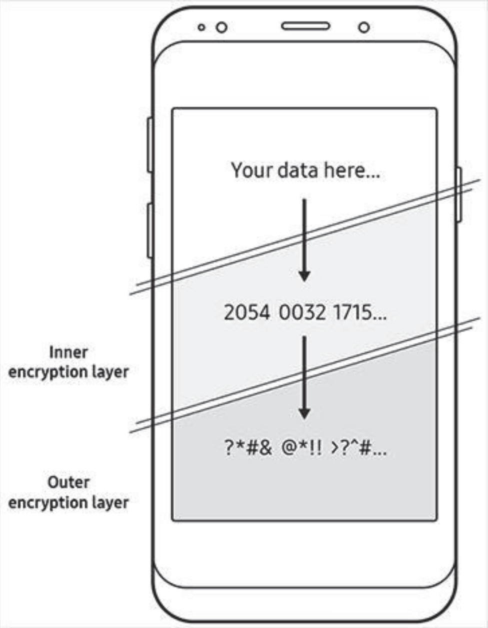
Knox DualDAR adds two separate layers of encryption, further meeting the requirements of classified deployments. Knox DualDAR secures all Workspace data on devices with two distinct levels of encryption. The solution also protects data by restricting apps from writing or saving data to the unencrypted space on the device. As the name implies, Knox DualDAR is based on two layers of data encryption. To fully understand how DualDAR works, we need to examine how the two layers of encryption within DualDAR work.

The DualDAR solution provides the following two separate layers of encryption and key generation. All data placed inside the Workspace is dually encrypted by both layers. Currently, DualDAR only secures data placed inside the Workspace or designated Work profile.

- **Outer layer**: The outer layer of the DualDAR solution is built on top of Android's FBE and enhanced by Samsung to meet MDFPP requirements. This layer is implemented through the SoC dedicated to flash storage encryption. In this context, the SoC could be Qualcomm Integrated Crypto Engine (ICE) or Exynos Flash Memory Protector (FMP). Data encryption at this layer is AES 256 XTS and file encryption keys are encrypted using AES-GCM 256.
- **Inner layer**: The inner layer of encryption is based on a framework that allows an independent third party to install a separate cryptographic module. If no third party module is installed, an separate inner layer of encryption is secured by a FIPS 140-2 certified cryptographic module included with the Samsung Knox framework.

DualDAR is supported on all devices with Knox version 3.3 or later and compatible with Android FBE. For more information on finding your Knox version, see the Prerequisite section on the DualDAR UEM integration page.

How DualDAR encryption works



DualDAR's inner and outer security layers are independent and protect all information stored in the Workspace when the device is in a powered off or unauthenticated state. Samsung Knox DualDAR leverages Android File Based Encryption (FBE) architecture.

On a FBE-enabled device, every device has the following two storage locations available to an app.

- **Credential Encrypted (CE) storage**: Default storage location and only available after a user has unlocked the device.
- **Device Encrypted (DE) storage**: Storage location available both during Direct Boot mode and after the user has unlocked the device.

From an app point of view, the DualDAR Workspace functions as CE storage. The Knox framework prevents apps from writing data to non-DualDAR protected DE storage. In some cases an app is aware of both CE and DE storage, and needs to write unclassified content to DE storage. In such cases, IT admins can whitelist that app to write to DE storage. This strict whitelist process ensures that no app can write sensitive or classified content to DE storage without explicit IT admin approval.

When the Workspace container is configured for DualDAR, the secured data is available as follows.

1. On a device that supports and is configured for DualDAR, access to app data inside the container is only available when the container is unlocked, that is when the user is actively using the container.
2. When the container—or device as a whole—is locked, the container encryption keys are evicted from memory.
3. In a data lock state, the Samsung device remains powered on but the user is locked out of both the Workspace and device. All sensitive data is protected in Credential Encrypted (CE) storage within the Workspace. CE storage is not available until the user provides both their device and Workspace credentials.

Unique advantages of Knox DualDAR

DualDAR encryption has the following significant advantages over traditional single layer encryption methods.

- **Mitigate risks of implementation flaws**: DualDAR reduces the likelihood of unauthorized data access by mitigating the risks that arise from vulnerabilities in a single encryption layer. While one of the many methods available for unauthorized data access may crack through a single layer of encryption, the chances are very low that such vulnerabilities are available on both layers of encryption.
- **Mitigate risks of password configuration flaws**: Both layers of encryption on a DualDAR configured device use separate and distinct authentication methods to allow access. This separation of authentication methods reduces the likelihood that a single misplaced or misconfigured password is exploited on both layers of data encryption at the same time. Two layers of encryption and two methods of authentication ensure that encrypted data remains protected even in the event of breach on one layer.
- **Provide access using strict security evaluation criteria**: DualDAR meets the standards laid out in the FIPS 140 certification requirements. Both the inner and outer layers use FIPS 140 certified cryptographic modules. GCM is used to encrypt the key while data is encrypted using XTS or CBC.
- **Ease of deployment**: DualDAR leverages the in-built Android FBE framework and builds additional layers of security on top of this framework. This solution is available on devices that use a Knox Workspace in PO mode as well as fully managed devices that include a PO mode. For more information on configuring this solution for your supported device, see the DualDAR architecture page.
- **Customize the second layer of encryption**: DualDAR allows IT admins to implement third party encryption solutions at the inner layer of encryption. This freedom of implementation means IT admins can use and configure any third party cryptographic modules, including solutions that meet FIPS 140 certification criteria.
- **Flexible deployment methods**: IT admins can implement and configure DualDAR on all kinds of devices, including BYOD and company-issued devices. Whether the device uses a Knox Workspace in PO mode or is a fully managed device that includes a PO mode, DualDAR is compatible with both models. This flexibility means IT admins can use this superior data security solution on a wide variety of devices within their enterprise.

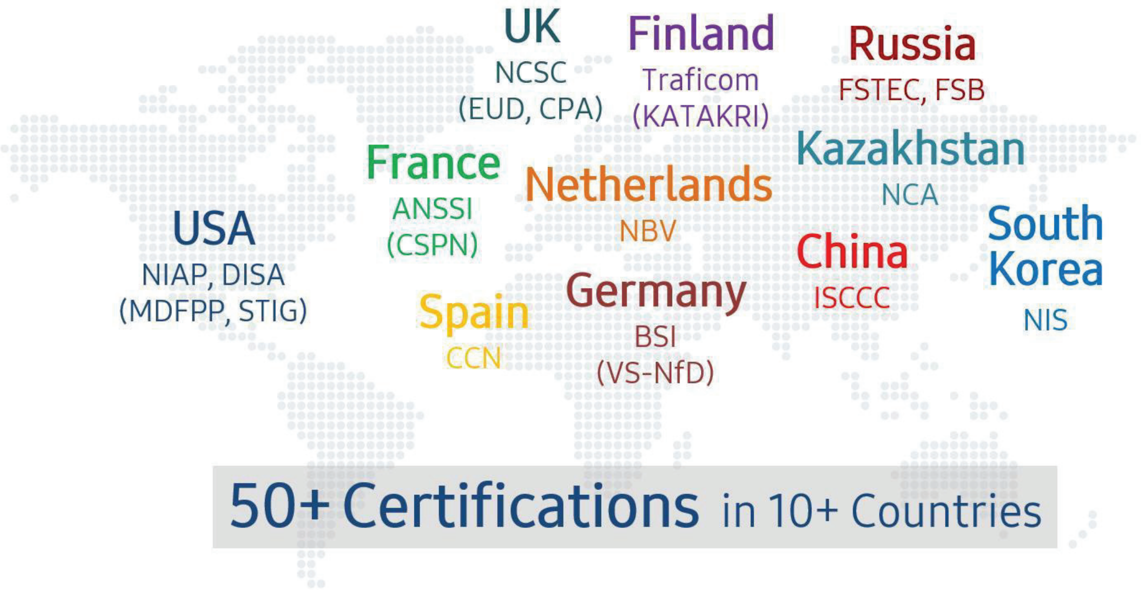
For more information on DualDAR and its unique design, see DualDAR architecture.

Appendix

Knox Certifications

The Knox Platform has successfully met the rigorous security requirements set by governments and major enterprises around the world, providing organizations with a trusted mobile security solution. The certifications acquired by the Knox Platform allow its mobile devices to be deployed in highly sensitive industries such as the military.

Samsung Knox continuously adds to its growing list of certifications for industries and agencies around the world. For more information on certifications and to review the latest list, see Knox certifications.



Unlike other mobile platforms, the Knox Platform is certified to have met the following countries' security requirements.

	USA	UK		Germany		France	Spain	Finland	Netherlands
	MDFPP	EUD	CPA	Endorsement	VS-NfD	CSPN	CCN	TRAFICOM	NCSA
Samsung	✔	✔	✔	✔	✔ pre-approved	✔	✔	✔	✔

Methodology

Certifications are granted by independent boards that use a specific set of hardware and software, for example, one certificate might be granted for the Galaxy S8 running Knox 3.0. These certifications must be renewed with each device and OS iteration to remain valid. Samsung remains dedicated to maintaining industry compliance and continues to grow and maintain our numerous certifications.

Security principles

Many of these certifications have a set of security principals that a device must uphold. Here are some examples of the security principles validated during certification.

- **Data-in-transit protection** — Does the device sufficiently protect data-in-transit?
  Yes - achieved with Advanced VPNs, Certificate Management, and Common Criteria mode.
- **Data-at-rest protection** — Does the device provide data that is encrypted by default? Is that data encrypted when the device is locked?
  Yes - achieved with the Knox Workspace and Sensitive Data Protection.
- **Authentication** — Does the device provide secure authentication methods?
  Yes - achieved with the Client Certificate Manager and user authentication methods that include biometrics.
- **Secure boot** — Does the device have mechanisms to ensure the boot up process is free from modification?
  Yes - achieved with a hardware-backed Root of Trust and Trusted Boot.
- **Platform integrity** — Does the device ensure the integrity of the platform? Can it query the integrity of the platform?
  Yes - achieved with the Real-Time Kernel Protection, Device Health Attestation, and Secure lockdown on tampering.
- **App sandboxing** — Does the device provide app sandboxing?
  Yes - achieved with through the Knox Workspace and SEAMS.
- **App Whitelisting** — Does the device allow app whitelisting and blacklisting?
  Yes - achieved with Advanced App Management.
- **Security policy enforcement** — Does the device allow the enforcement of security policies? Can they take precedence over user activities?
  Yes - achieved with a full complement of EMM policies built on a Knox SDK offering over 1500 APIs.
- **External interface protection** — Does the device allow control over external peripherals such as Bluetooth, USB, and NFC?
  Yes - achieved with Granular Device Management.
- **Device Update Policy** — Can the device provide deliberate OS updates that match an organizations evolving needs?
  Yes - achieved with Device Software Update Management.
- **Event collection for enterprise analysis** — Does the device allow the collection, and subsequent audit, of business data?
  Yes - achieved with Audit Logs.
- **Incident Response** — Can the device be managed if it is lost, stolen or damaged?
  Yes - achieved with custom lockscreen info, remote data wipe, auto-wipe after a number of unsuccessful log-in attempts, and remote factory reset.

What does this mean to you? You can rest easy knowing that Samsung Knox's holistic security platform is compliant with the highest security requirements and standards. Samsung Knox devices are built from the ground up to secure your organization's apps and data, providing robust integration with existing IT infrastructure and ensuring there are no functional or security gaps in your deployment.

Common Criteria Mode

Knox supports advanced device configurations tailored to the defense industry. A single Knox setting can apply many of the settings needed to put the device into a compliant state. This setting, called Common Criteria Mode or CC Mode, helps simplify the task of correctly configuring a device for deployments that must meet defense-grade security requirements. The Common Criteria for Information Technology Security Evaluation, commonly referred to as Common Criteria, is an internationally-recognized standard for defining security objectives of information technology products and for evaluating vendor compliance with these objectives. A number of Governments use Common Criteria as the basis for their own certification schemes.

Select Samsung Galaxy devices with the Knox Platform embedded received Common Criteria (CC) certification. The current CC certification targets the new Mobile Device Fundamentals Protection Profile (MDFPP) of the National Information Assurance Partnership (NIAP), which addresses the security requirements of mobile devices for use in enterprise. Samsung Knox is approved by the United States Government as the first NIAP-validated consumer mobile devices to handle the full range of classified information.

An IT admin can enable the device to be placed into the Common Criteria configuration. When enabled, the device:

- Blocks bootloader download mode, the manual method for software updates
- Mandates additional key zeroization on key deletion
- Prevents non-authenticated Bluetooth connections
- Requires that FOTA updates have a 2048-bit RSA-PSS signature
- Uses many other security settings

While other optional configuration steps are still recommended on top of Common Criteria Mode, the value is clear: simplifying the correct configuration of endpoints for high-security deployments saves time and prevents mistakes that can lead to misconfigurations and added security risks.

More information

Refer to the following Knowledge Base Articles for details about:

- Common Criteria Mode, supported Samsung devices, and test APKs
- Common Criteria evaluation, by Android version

EXHIBIT 6

RSA LABORATORIES

PKCS #11 v2.20 Amendment 1
PKCS #11 Mechanisms for One-Time Password Tokens

RSA Laboratories

December 27, 2005

TABLE OF CONTENTS

1 INTRODUCTION 2

1.1 SCOPE 2

1.2 BACKGROUND 2

1.3 DOCUMENT ORGANIZATION 3

2 USAGE OVERVIEW 3

2.1 CASE 1: GENERATION OF OTP VALUES 4

2.2 CASE 2: VERIFICATION OF PROVIDED OTP VALUES 5

2.3 CASE 3: GENERATION OF OTP KEYS 6

3 OTP OBJECTS 6

3.1 KEY OBJECTS 6

4 OTP-RELATED NOTIFICATIONS 9

5 OTP MECHANISMS 9

5.1 OTP MECHANISM PARAMETERS 10

- ♦ *CK_PARAM_TYPE 10*
- ♦ *CK_OTP_PARAM; CK_OTP_PARAM_PTR 12*
- ♦ *CK_OTP_PARAMS; CK_OTP_PARAMS_PTR 13*
- ♦ *CK_OTP_SIGNATURE_INFO, CK_OTP_SIGNATURE_INFO_PTR 13*

5.2 RSA SECURID 14

5.2.1 RSA SecurID secret key objects 14

5.2.2 RSA SecurID key generation 15

5.2.3 RSA SecurID OTP generation and validation 16

5.2.4 Return values 16

5.3 OATH HOTP 16

5.3.1 OATH HOTP secret key objects 16

5.3.2 HOTP key generation 17

5.3.3 HOTP OTP generation and validation 17

5.4 ACTIVIDENTITY ACTI 18

5.4.1 ACTI secret key objects 18

5.4.2 ACTI key generation 19

5.4.3 ACTI OTP generation and validation 19

A. MANIFEST CONSTANTS 20

A.1 OBJECT CLASSES 20

A.2 KEY TYPES 20

A.3 MECHANISMS 20

A.4 ATTRIBUTES 20

A.5 ATTRIBUTE CONSTANTS 20

A.6 OTHER CONSTANTS 21

A.7 NOTIFICATIONS 21

A.8 RETURN VALUES 21

B. EXAMPLE CODE 21

B.1 DISCLAIMER CONCERNING SAMPLE CODE 21

B.2 OTP RETRIEVAL 21

B.3 USER-FRIENDLY MODE OTP TOKEN 25

B.4 OTP VERIFICATION 26

C. INTELLECTUAL PROPERTY CONSIDERATIONS 27

D. REFERENCES 28

E. ABOUT OTPS 28

1 Introduction

1.1 Scope

This document is an amendment to PKCS #11 v.20 [1] and describes general PKCS #11 objects, procedures and mechanisms that can be used to retrieve and verify one-time passwords (OTPs) generated by OTP tokens.

1.2 Background

A One-Time Password (OTP) token may be a handheld hardware device, a hardware device connected to a personal computer through an electronic interface such as USB, or a software module resident on a personal computer, which generates one-time passwords that may be used to authenticate a user towards some service. Increasingly, these tokens work in a connected fashion, enabling programmatic retrieval of their OTP values. To meet the needs of applications wishing to access these connected OTP tokens in an interoperable manner, this document extends PKCS #11 [1] to better support these tokens, easing the task for vendors of OTP-consuming applications, and enabling a better user experience.

This document adds basic support of One-Time Password (OTP) tokens to PKCS #11 by defining a common OTP key type with an extensible set of attributes and by describing how PKCS #11 functions can be used to retrieve and verify OTP values generated by an OTP token. It also describes an OTP key generation mechanism that may be used to execute on-token key generation.

Building on the OTP framework, the document specifies the PKCS #11 RSA SecurID™ OTP mechanisms[1], the OATH HOTP mechanisms[2], and the ActivIdentity ACTI mechanisms. Additional mechanisms may be defined separately to support other types of OTP tokens.

A Cryptoki library supporting OTP tokens and the PKCS #11 v2.20 extensions defined herein may also support existing PKCS #11 cryptographic tokens. It is also envisioned that certain tokens will offer both OTP functionality and traditional cryptographic token functionalities such as encryption, decryption, etc.

1.3 Document organization

The organization of this document is as follows:

- Section 1 is an introduction.
- Section 2 provides an overview description of the support for OTP tokens in PKCS #11 defined herein.
- Section 3 defines the new OTP key object type and its attributes.
- Section 4 defines a new OTP-related notification.
- Section 5 defines specific OTP mechanisms.
- Appendix A collects defined PKCS #11 constants.
- Appendix B provides example usages of the OTP mechanisms.
- Appendices C, D, and E cover intellectual property issues, give references to other publications and standards, and provide general information about the One-Time Password Specifications.

2 Usage overview

OTP tokens represented as PKCS #11 mechanisms may be used in a variety of ways. The usage cases can be categorized according to the type of sought functionality.

[1] RSA SecurID® two-factor authentication is a symmetric authentication method which is patented by RSA Security. A user authenticates by submitting a one-time password (OTP), or PASSCODE value generated by an RSA SecurID token. The RSA SecurID token may be a handheld hardware device, a hardware device connected to a personal computer through an electronic interface such as USB, or a software module resident on the personal computer.

[2] The HOTP algorithm is work in progress, currently defined in the IETF draft http://www.ietf.org/internet-drafts/draft-mraihi-oath-hmac-otp-04.txt developed by the Open Authentication initiative (http://www.openauthentication.org).

2.1 Case 1: Generation of OTP values

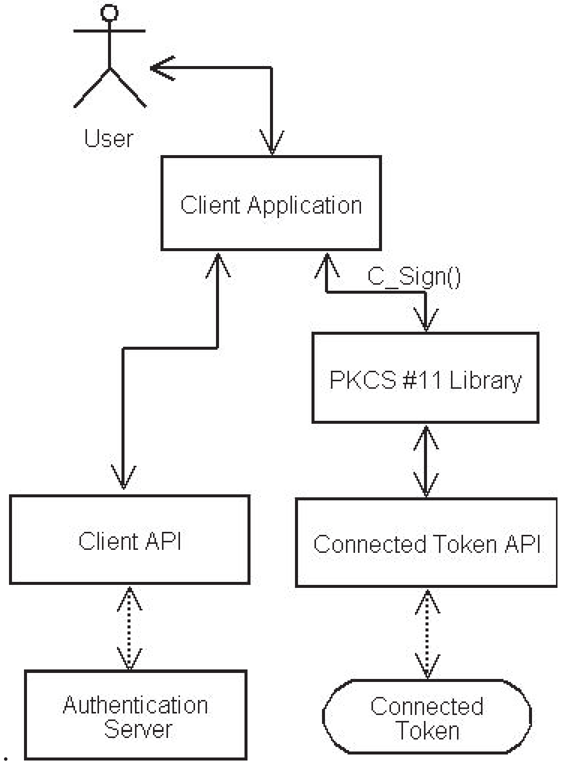


Figure 1: Retrieving OTP values through C_Sign

Figure 1 shows an integration of PKCS #11 into an application that needs to authenticate users holding OTP tokens. In this particular example, a connected hardware token is used, but a software token is equally possible. The application invokes **C_Sign** to retrieve the OTP value from the token. In the example, the application then passes the retrieved OTP value to a client API that sends it via the network to an authentication server. The client API may implement a standard authentication protocol such as RADIUS [2] or EAP [3], or a proprietary protocol such as that used by RSA Security's ACE/Agent® software.

2.2 Case 2: Verification of provided OTP values

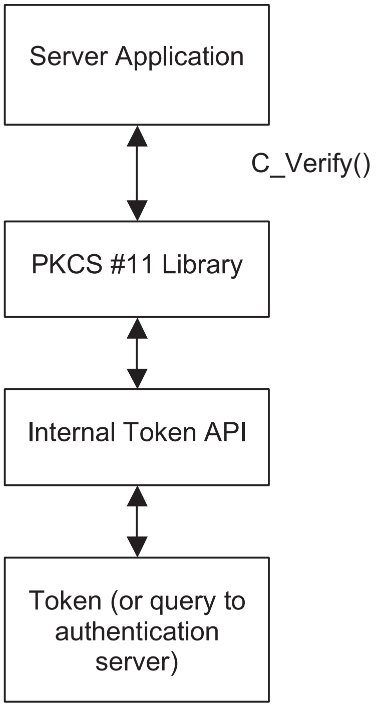


Figure 2: Server-side verification of OTP values

Figure 2 illustrates the server-side equivalent of the scenario depicted in Figure 1. In this case, a server application invokes **C_Verify** with the received OTP value as the signature value to be verified.

2.3 Case 3: Generation of OTP keys

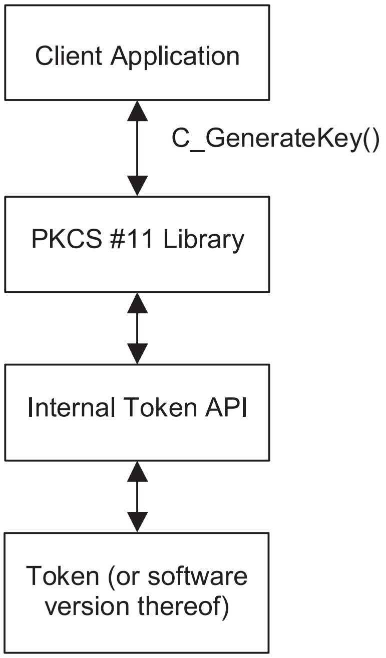


Figure 3: Generation of an OTP key

Figure 3 shows an integration of PKCS #11 into an application that generates OTP keys. The application invokes **C_GenerateKey** to generate an OTP key of a particular type on the token. The key may subsequently be used as a basis to generate OTP values.

3 OTP objects

3.1 Key objects

OTP key objects (object class **CKO_OTP_KEY**) hold secret keys used by OTP tokens. The following table defines the attributes common to all OTP keys, in addition to the attributes defined for secret keys, all of which are inherited by this class:

Table 1: Common OTP key attributes

Attribute	Data type	Meaning

CKA_OTP_FORMAT	CK_ULONG	Format of OTP values produced with this key:
		CK_OTP_FORMAT_DECIMAL = Decimal (default) (UTF8-encoded)
		CK_OTP_FORMAT_HEXADECIMAL = Hexadecimal (UTF8-encoded)
		CK_OTP_FORMAT_ALPHANUMERIC = Alphanumeric (UTF8-encoded)
		CK_OTP_FORMAT_BINARY = Only binary values.

CKA_OTP_LENGTH[9]	CK_ULONG	Default length of OTP values (in the CKA_OTP_FORMAT) produced with this key.
CKA_OTP_USER_FRIENDLY_MODE[9]	CK_BBOOL	Set to CK_TRUE when the token is capable of returning OTPs suitable for human consumption. See the description of CKF_USER_FRIENDLY_OTP below.
CKA_OTP _CHALLENGE_REQUIREMENT[9]	CK_ULONG	Parameter requirements when generating or verifying OTP values with this key: CK_OTP_PARAM_MANDATORY = A challenge must be supplied. CK_OTP_PARAM_OPTIONAL = A challenge may be supplied but need not be. CK_OTP_PARAM_IGNORED = A challenge, if supplied, will be ignored.
CKA_OTP_TIME_REQUIREMENT[9]	CK_ULONG	Parameter requirements when generating or verifying OTP values with this key: CK_OTP_PARAM_MANDATORY = A time value must be supplied. CK_OTP_PARAM_OPTIONAL = A time value may be supplied but need not be. CK_OTP_PARAM_IGNORED = A time value, if supplied, will be ignored.
CKA_OTP_COUNTER_REQUIREMENT[9]	CK_ULONG	Parameter requirements when generating or verifying OTP values with this key: CK_OTP_PARAM_MANDATORY = A counter value must be supplied. CK_OTP_PARAM_OPTIONAL = A

Attribute	Data type	Meaning
		counter value may be supplied but need not be. CK_OTP_PARAM_IGNORED = A counter value, if supplied, will be ignored.
CKA_OTP_PIN_REQUIREMENT[9]	CK_ULONG	Parameter requirements when generating or verifying OTP values with this key: CK_OTP_PARAM_MANDATORY = A PIN value must be supplied. CK_OTP_PARAM_OPTIONAL = A PIN value may be supplied but need not be (if not supplied, then library will be responsible for collecting it) CK_OTP_PARAM_IGNORED = A PIN value, if supplied, will be ignored.
CKA_OTP_COUNTER	Byte array	Value of the associated internal counter. Default value is empty (i.e. *ulValueLen* = 0).
CKA_OTP_TIME	RFC 2279 string	Value of the associated internal UTC time in the form YYYYMMDDhhmmss. Default value is empty (i.e. *ulValueLen*= 0).
CKA_OTP_USER_IDENTIFIER	RFC 2279 string	Text string that identifies a user associated with the OTP key (may be used to enhance the user experience). Default value is empty (i.e. *ulValueLen* = 0).
CKA_OTP_SERVICE_IDENTIFIER	RFC 2279 string	Text string that identifies a service that may validate OTPs generated by this key. Default value is empty (i.e. *ulValueLen* = 0).
CKA_OTP_SERVICE_LOGO	Byte array	Logotype image that identifies a service that may validate OTPs generated by this key. Default value is empty (i.e. *ulValueLen* = 0).
CKA_OTP_SERVICE_LOGO_TYPE	RFC 2279 string	MIME type of the CKA_OTP_SERVICE_LOGO attribute value. Default value is empty (i.e. *ulValueLen* = 0).
CKA_VALUE[1, 4, 6, 7]	Byte array	Value of the key.
CKA_VALUE_LEN[2, 3]	CK_ULONG	Length in bytes of key value.

Refer to Table 15 in [1] for table footnotes.

Note: A Cryptoki library may support PIN-code caching in order to reduce user interactions. An OTP-PKCS #11 application should therefore always consult the state of the CKA_OTP_PIN_REQUIREMENT attribute before each call to **C_SignInit**, as the value of this attribute may change dynamically.

For OTP tokens with multiple keys, the keys may be enumerated using **C_FindObjects**. The **CKA_OTP_SERVICE_IDENTIFIER** and/or the **CKA_OTP_SERVICE_LOGO** attribute may be used to distinguish between keys. The actual choice of key for a particular operation is however application-specific and beyond the scope of this document.

For all OTP keys, the CKA_ALLOWED_MECHANISMS attribute should be set in accordance with [1], Table 27.

4 OTP-related notifications

This document extends the set of defined notifications as follows:

CKN_OTP_CHANGED Cryptoki is informing the application that the OTP for a key on a connected token just changed. This notification is particularly useful when applications wish to display the current OTP value for time-based mechanisms.

5 OTP mechanisms

The following table shows, for the OTP mechanisms defined in this document, their support by different cryptographic operations. For any particular token, of course, a particular operation may well support only a subset of the mechanisms listed. There is also no guarantee that a token that supports one mechanism for some operation supports any other mechanism for any other operation (or even supports that same mechanism for any other operation).

Table 2: OTP mechanisms vs. applicable functions

Mechanism	Functions						
	Encrypt & Decrypt	Sign & Verify	SR & VR[1]	Digest	Gen. Key/ Key Pair	Wrap & Unwrap	Derive
CKM_SECURID_KEY_GEN					✓		
CKM_SECURID		✓					
CKM_HOTP_KEY_GEN					✓		
CKM_HOTP		✓					
CKM_ACTI_KEY_GEN					✓		
CKM_ACTI		✓					

The remainder of this section will present in detail the OTP mechanisms and the parameters that are supplied to them.

5.1 OTP mechanism parameters

♦ CK_PARAM_TYPE

CK_PARAM_TYPE is a value that identifies an OTP parameter type. It is defined as follows:

```
typedef CK_ULONG CK_PARAM_TYPE;
```

The following **CK_PARAM_TYPE** types are defined:

Table 3: OTP parameter types

Parameter	**Data type**	**Meaning**
CK_OTP_PIN	RFC 2279 string	A UTF8 string containing a PIN for use when computing or verifying PIN-based OTP values.
CK_OTP_CHALLENGE	Byte array	Challenge to use when computing or verifying challenge-based OTP values.
CK_OTP_TIME	RFC 2279 string	UTC time value in the form YYYYMMDDhhmmss to use when computing or verifying time-based OTP values.
CK_OTP_COUNTER	Byte array	Counter value to use when computing or verifying counter-based OTP values.
CK_OTP_FLAGS	CK_FLAGS	Bit flags indicating the characteristics of the sought OTP as defined below.
CK_OTP_OUTPUT_LENGTH	CK_ULONG	Desired output length (overrides any default value). A Cryptoki library will return CKR_MECHANISM_PARAM_INVALID if a provided length value is not supported.
CK_OTP_FORMAT	CK_ULONG	Returned OTP format (allowed values are the same as for CKA_OTP_FORMAT). This parameter is only intended for **C_Sign** output, see below. When not present, the returned OTP format will be the same as the value of the CKA_OTP_FORMAT attribute for the key in question.
CK_OTP_VALUE	Byte array	An actual OTP value. This parameter type is intended for **C_Sign** output, see below.

The following table defines the possible values for the CK_OTP_FLAGS type:

Table 4: OTP Mechanism Flags

Bit flag	**Mask**	**Meaning**
CKF_NEXT_OTP	0x00000001	True (i.e. set) if the OTP computation shall be for the next OTP, rather than the current one (current being interpreted in the context of the algorithm, e.g. for the current counter value or current time window). A Cryptoki library shall return CKR_MECHANISM_PARAM_INVALID if the CKF_NEXT_OTP flag is set and the OTP mechanism in question does not support the concept of "next" OTP or the library is not capable of generating the next OTP[3].
CKF_EXCLUDE_TIME	0x00000002	True (i.e. set) if the OTP computation must not include a time value. Will have an effect only on mechanisms that do include a time value in the OTP computation and then only if the mechanism (and token) allows exclusion of this value. A Cryptoki library shall return CKR_MECHANISM_PARAM_INVALID if exclusion of the value is not allowed.
CKF_EXCLUDE_COUNTER	0x00000004	True (i.e. set) if the OTP computation must not include a counter value. Will have an effect only on mechanisms that do include a counter value in the OTP computation and then only if the mechanism (and token) allows exclusion of this value. A Cryptoki library shall return CKR_MECHANISM_PARAM_INVALID if exclusion of the value is not allowed.
CKF_EXCLUDE_CHALLENGE	0x00000008	True (i.e. set) if the OTP computation must not include a challenge. Will have an effect only on mechanisms that do include a challenge in the OTP computation and then only if the mechanism (and token) allows exclusion of this value. A Cryptoki library shall return CKR_MECHANISM_PARAM_INVALID if exclusion of the value is not allowed.

[3] Applications that may need to retrieve the next OTP should be prepared to handle this situation. For example, an application could store the OTP value returned by C_Sign so that, if a next OTP is required, it can compare it to the OTP value returned by subsequent calls to C_Sign should it turn out that the library does not support the CKF_NEXT_OTP flag.

Bit flag	Mask	Meaning
CKF_EXCLUDE_PIN	0x00000010	True (i.e. set) if the OTP computation must not include a PIN value. Will have an effect only on mechanisms that do include a PIN in the OTP computation and then only if the mechanism (and token) allows exclusion of this value. A Cryptoki library shall return CKR_MECHANISM_PARAM_INVALID if exclusion of the value is not allowed.
CKF_USER_FRIENDLY_OTP	0x00000020	True (i.e. set) if the OTP returned shall be in a form suitable for human consumption. If this flag is set, and the call is successful, then the returned CK_OTP_VALUE shall be a UTF8-encoded printable string. A Cryptoki library shall return CKR_MECHANISM_PARAM_INVALID if this flag is set when CKA_OTP_USER_FRIENDLY_MODE for the key in question is CK_FALSE.

Note: Even if CKA_OTP_FORMAT is not set to CK_OTP_FORMAT_BINARY, then there may still be value in setting the CKF_USER_FRIENDLY flag (assuming CKA_USER_FRIENDLY_MODE is CK_TRUE, of course) if the intent is for a human to read the generated OTP value, since it may become shorter or otherwise better suited for a user. Applications that do not intend to provide a returned OTP value to a user should not set the CKF_USER_FRIENDLY_OTP flag.

♦ CK_OTP_PARAM; CK_OTP_PARAM_PTR

CK_OTP_PARAM is a structure that includes the type, value, and length of an OTP parameter. It is defined as follows:

```
typedef struct CK_OTP_PARAM {
   CK_PARAM_TYPE type;
   CK_VOID_PTR pValue;
   CK_ULONG ulValueLen;
} CK_OTP_PARAM;
```

The fields of the structure have the following meanings:

type	the parameter type
pValue	pointer to the value of the parameter
ulValueLen	length in bytes of the value

If a parameter has no value, then *ulValueLen* = 0, and the value of *pValue* is irrelevant. Note that *pValue* is a "void" pointer, facilitating the passing of arbitrary values. Both the application and the Cryptoki library must ensure that the pointer can be safely cast to the expected type (*i.e.*, without word-alignment errors).

CK_OTP_PARAM_PTR is a pointer to a **CK_OTP_PARAM.**

♦ CK_OTP_PARAMS; CK_OTP_PARAMS_PTR

CK_OTP_PARAMS is a structure that is used to provide parameters for OTP mechanisms in a generic fashion. It is defined as follows:

```
typedef struct CK_OTP_PARAMS {
   CK_OTP_PARAM_PTR pParams;
   CK_ULONG ulCount;
} CK_OTP_PARAMS;
```

The fields of the structure have the following meanings:

pParams	pointer to an array of OTP parameters
ulCount	the number of parameters in the array

CK_OTP_PARAMS_PTR is a pointer to a **CK_OTP_PARAMS**.

When calling **C_SignInit** or **C_VerifyInit** with a mechanism that takes a **CK_OTP_PARAMS** structure as a parameter, the **CK_OTP_PARAMS** structure shall be populated in accordance with the **CKA_OTP_*X*_REQUIREMENT** key attributes for the identified key, where ***X*** is **PIN**, **CHALLENGE**, **TIME**, or **COUNTER**.

For example, if **CKA_OTP_TIME_REQUIREMENT** = CK_OTP_PARAM_MANDATORY, then the **CK_OTP_TIME** parameter shall be present. If **CKA_OTP_TIME_REQUIREMENT** = CK_OTP_PARAM_OPTIONAL, then a **CK_OTP_TIME** parameter may be present. If it is not present, then the library may collect it (during the **C_Sign** call). If **CKA_OTP_TIME_REQUIREMENT** = CK_OTP_PARAM_IGNORED, then a provided **CK_OTP_TIME** parameter will always be ignored. Additionally, a provided **CK_OTP_TIME** parameter will always be ignored if CKF_EXCLUDE_TIME is set in a **CK_OTP_FLAGS** parameter. Similarly, if this flag is set, a library will not attempt to collect the value itself, and it will also instruct the token not to make use of any internal value, subject to token policies. It is an error (**CKR_MECHANISM_PARAM_INVALID**) to set the CKF_EXCLUDE_TIME flag when the **CKA_TIME_REQUIREMENT** attribute is CK_OTP_PARAM_MANDATORY.

The above discussion holds for all **CKA_OTP_*X*_REQUIREMENT** attributes (*i.e.*, **CKA_OTP_PIN_REQUIREMENT**, **CKA_OTP_CHALLENGE_REQURIEMENT**, **CKA_OTP_COUNTER_REQUIREMENT**, **CKA_OTP_TIME_REQUIREMENT**). A library may set a particular **CKA_OTP_*X*_REQUIREMENT** attribute to CK_OTP_PARAM_OPTIONAL even if it is required by the mechanism as long as the token (or the library itself) has the capability of providing the value to the computation. One example of this is a token with an on-board clock.

In addition, applications may use the **CK_OTP_FLAGS**, the **CK_OTP_OUTPUT_FORMAT** and the **CK_OUTPUT_LENGTH** parameters to set additional parameters.

♦ CK_OTP_SIGNATURE_INFO, CK_OTP_SIGNATURE_INFO_PTR

CK_OTP_SIGNATURE_INFO is a structure that is returned by all OTP mechanisms in successful calls to **C_Sign** (**C_SignFinal**). The structure informs applications of actual parameter values used in particular OTP computations in addition to the OTP value itself. It is used by all mechanisms for which the key belongs to the class CKO_OTP_KEY and is defined as follows:

```
typedef struct CK_OTP_SIGNATURE_INFO {
   CK_OTP_PARAM_PTR pParams;
   CK_ULONG ulCount;
} CK_OTP_SIGNATURE_INFO;
```

The fields of the structure have the following meanings:

pParams pointer to an array of OTP parameter values

ulCount the number of parameters in the array

After successful calls to **C_Sign** or **C_SignFinal** with an OTP mechanism, the *pSignature* parameter will be set to point to a **CK_OTP_SIGNATURE_INFO** structure. One of the parameters in this structure will be the OTP value itself, identified with the **CK_OTP_VALUE** tag. Other parameters may be present for informational purposes, e.g. the actual time used in the OTP calculation. In order to simplify OTP validations, authentication protocols may permit authenticating parties to send some or all of these parameters in addition to OTP values themselves. Applications should therefore check for their presence in returned **CK_OTP_SIGNATURE_INFO** values whenever such circumstances apply.

Since **C_Sign** and **C_SignFinal** follows the convention described in Section 11.2 of [1] on producing output, a call to **C_Sign** (or **C_SignFinal**) with *pSignature* set to NULL_PTR will return (in the *pulSignatureLen* parameter) the required number of bytes to hold the **CK_OTP_SIGNATURE_INFO** structure *as well as all the data in all its **CK_OTP_PARAM** components*. If an application allocates a memory block based on this information, it shall therefore not subsequently de-allocate components of such a received value but rather de-allocate the complete **CK_OTP_PARAMS** structure itself. A Cryptoki library that is called with a non-NULL *pSignature* pointer will assume that it points to a *contiguous* memory block of the size indicated by the *pulSignatureLen* parameter.

When verifying an OTP value using an OTP mechanism, *pSignature* shall be set to the OTP value itself, e.g. the value of the **CK_OTP_VALUE** component of a **CK_OTP_PARAMS** structure returned by a call to **C_Sign**. The **CK_OTP_PARAMS** value supplied in the **C_VerifyInit** call sets the values to use in the verification operation.

CK_OTP_SIGNATURE_INFO_PTR points to a **CK_OTP_SIGNATURE_INFO.**

5.2 RSA SecurID

5.2.1 RSA SecurID secret key objects

RSA SecurID secret key objects (object class **CKO_OTP_KEY,** key type **CKK_SECURID**) hold RSA SecurID secret keys. The following table defines the RSA SecurID secret key object attributes, in addition to the common attributes defined for this object class:

Table 5: RSA SecurID secret key object attributes

Attribute	Data type	Meaning
CKA_OTP_TIME_INTERVAL[1]	CK_ULONG	Interval between OTP values produced with this key, in seconds. Default is 60.

Refer to Table 15 in [1] for table footnotes.

The following is a sample template for creating an RSA SecurID secret key object:

```
CK_OBJECT_CLASS class = CKO_OTP_KEY;
CK_KEY_TYPE keyType = CKK_SECURID;
CK_DATE endDate = {...};
CK_UTF8CHAR label[] = "RSA SecurID secret key object";
CK_BYTE keyId[]= {...};
CK_ULONG outputFormat = CK_OTP_FORMAT_DECIMAL;
CK_ULONG outputLength = 6;
CK_ULONG needPIN = CK_OTP_PARAM_MANDATORY;
CK_ULONG timeInterval = 60;
CK_BYTE value[] = {...};
CK_BBOOL true = CK_TRUE;
CK_ATTRIBUTE template[] = {
   {CKA_CLASS, &class, sizeof(class)},
   {CKA_KEY_TYPE, &keyType, sizeof(keyType)},
   {CKA_END_DATE, &endDate, sizeof(endDate)},
   {CKA_TOKEN, &true, sizeof(true)},
   {CKA_SENSITIVE, &true, sizeof(true)},
   {CKA_LABEL, label, sizeof(label)-1},
   {CKA_SIGN, &true, sizeof(true)},
   {CKA_VERIFY, &true, sizeof(true)},
   {CKA_ID, keyId, sizeof(keyId)},
   {CKA_OTP_FORMAT, &outputFormat,
        sizeof(outputFormat)},
   {CKA_OTP_LENGTH, &outputLength,
        sizeof(outputLength)},
   {CKA_OTP_PIN_REQUIREMENT, &needPIN, sizeof(needPIN)},
   {CKA_OTP_TIME_INTERVAL, &timeInterval,
        sizeof(timeInterval)},
   {CKA_VALUE, value, sizeof(value)}
};
```

5.2.2 RSA SecurID key generation

The RSA SecurID key generation mechanism, denoted **CKM_SECURID_KEY_GEN**, is a key generation mechanism for the RSA SecurID algorithm.

It does not have a parameter.

The mechanism generates RSA SecurID keys with a particular set of attributes as specified in the template for the key.

The mechanism contributes at least the **CKA_CLASS**, **CKA_KEY_TYPE**, **CKA_VALUE_LEN**, and **CKA_VALUE** attributes to the new key. Other attributes supported by the RSA SecurID key type may be specified in the template for the key, or else are assigned default initial values

For this mechanism, the *ulMinKeySize* and *ulMaxKeySize* fields of the **CK_MECHANISM_INFO** structure specify the supported range of SecurID key sizes, in bytes.

5.2.3 RSA SecurID OTP generation and validation

CKM_SECURID is the mechanism for the retrieval and verification of RSA SecurID OTP values.

The mechanism takes a pointer to a **CK_OTP_PARAMS** structure as a parameter.

When signing or verifying using the **CKM_SECURID** mechanism, *pData* shall be set to NULL_PTR and *ulDataLen* shall be set to 0.

5.2.4 Return values

Support for the **CKM_SECURID** mechanism extends the set of return values for **C_Verify** with the following values:

- CKR_NEW_PIN_MODE: The supplied OTP was not accepted and the library requests a new OTP computed using a new PIN. The new PIN is set through means out of scope for this document.
- CKR_NEXT_OTP: The supplied OTP was correct but indicated a larger than normal drift in the token's internal state (e.g. clock, counter). To ensure this was not due to a temporary problem, the application should provide the next one-time password to the library for verification.

5.3 OATH HOTP

5.3.1 OATH HOTP secret key objects

HOTP secret key objects (object class **CKO_OTP_KEY,** key type **CKK_HOTP**) hold generic secret keys and associated counter values.

The **CKA_OTP_COUNTER** value may be set at key generation; however, some tokens may set it to a fixed initial value. Depending on the token’s security policy, this value may not be modified and/or may not be revealed if the object has its **CKA_SENSITIVE** attribute set to CK_TRUE or its **CKA_EXTRACTABLE** attribute set to CK_FALSE.

For HOTP keys, the **CKA_OTP_COUNTER** value shall be an 8 bytes unsigned integer in big endian (i.e. network byte order) form. The same holds true for a **CK_OTP_COUNTER** value in a **CK_OTP_PARAM** structure.

The following is a sample template for creating a HOTP secret key object:

```
CK_OBJECT_CLASS class = CKO_OTP_KEY;
CK_KEY_TYPE keyType = CKK_HOTP;
```

```
CK_UTF8CHAR label[] = "HOTP secret key object";
CK_BYTE keyId[]= {...};
CK_ULONG outputFormat = CK_OTP_FORMAT_DECIMAL;
CK_ULONG outputLength = 6;
CK_DATE endDate = {...};
CK_BYTE counterValue[8] = {0};
CK_BYTE value[] = {...};
CK_BBOOL true = CK_TRUE;
CK_ATTRIBUTE template[] = {
   {CKA_CLASS, &class, sizeof(class)},
   {CKA_KEY_TYPE, &keyType, sizeof(keyType)},
   {CKA_END_DATE, &endDate, sizeof(endDate)},
   {CKA_TOKEN, &true, sizeof(true)},
   {CKA_SENSITIVE, &true, sizeof(true)},
   {CKA_LABEL, label, sizeof(label)-1},
   {CKA_SIGN, &true, sizeof(true)},
   {CKA_VERIFY, &true, sizeof(true)},
   {CKA_ID, keyId, sizeof(keyId)},
   {CKA_OTP_FORMAT, &outputFormat,
        sizeof(outputFormat)},
   {CKA_OTP_LENGTH, &outputLength,
        sizeof(outputLength)},
   {CKA_OTP_COUNTER, counterValue,
        sizeof(counterValue)},
   {CKA_VALUE, value, sizeof(value)}
};
```

5.3.2 HOTP key generation

The HOTP key generation mechanism, denoted **CKM_HOTP_KEY_GEN**, is a key generation mechanism for the HOTP algorithm.

It does not have a parameter.

The mechanism generates HOTP keys with a particular set of attributes as specified in the template for the key.

The mechanism contributes at least the **CKA_CLASS**, **CKA_KEY_TYPE**, **CKA_OTP_COUNTER**, **CKA_VALUE** and **CKA_VALUE_LEN** attributes to the new key. Other attributes supported by the HOTP key type may be specified in the template for the key, or else are assigned default initial values.

For this mechanism, the *ulMinKeySize* and *ulMaxKeySize* fields of the **CK_MECHANISM_INFO** structure specify the supported range of HOTP key sizes, in bytes.

5.3.3 HOTP OTP generation and validation

CKM_HOTP is the mechanism for the retrieval and verification of HOTP OTP values based on the current internal counter, or a provided counter.

The mechanism takes a pointer to a **CK_OTP_PARAMS** structure as a parameter.

As for the **CKM_SECURID** mechanism, when signing or verifying using the **CKM_HOTP** mechanism, *pData* shall be set to NULL_PTR and *ulDataLen* shall be set to 0.

For verify operations, the counter value **CK_OTP_COUNTER** must be provided as a **CK_OTP_PARAM** parameter to **C_VerifyInit**. When verifying an OTP value using the **CKM_HOTP** mechanism, *pSignature* shall be set to the OTP value itself, e.g. the value of the **CK_OTP_VALUE** component of a **CK_OTP_PARAMS** structure in the case of an earlier call to **C_Sign**.

5.4 ActivIdentity ACTI

5.4.1 ACTI secret key objects

ACTI secret key objects (object class **CKO_OTP_KEY,** key type **CKK_ACTI**) hold ActivIdentity ACTI secret keys.

For ACTI keys, the **CKA_OTP_COUNTER** value shall be an 8 bytes unsigned integer in big endian (i.e. network byte order) form. The same holds true for the **CK_OTP_COUNTER** value in the **CK_OTP_PARAM** structure.

The **CKA_OTP_COUNTER** value may be set at key generation; however, some tokens may set it to a fixed initial value. Depending on the token's security policy, this value may not be modified and/or may not be revealed if the object has its **CKA_SENSITIVE** attribute set to CK_TRUE or its **CKA_EXTRACTABLE** attribute set to CK_FALSE.

The **CKA_OTP_TIME** value may be set at key generation; however, some tokens may set it to a fixed initial value. Depending on the token's security policy, this value may not be modified and/or may not be revealed if the object has its **CKA_SENSITIVE** attribute set to CK_TRUE or its **CKA_EXTRACTABLE** attribute set to CK_FALSE.

The following is a sample template for creating an ACTI secret key object:

```
CK_OBJECT_CLASS class = CKO_OTP_KEY;
CK_KEY_TYPE keyType = CKK_ACTI;
CK_UTF8CHAR label[] = “ACTI secret key object”;
CK_BYTE keyId[]= {...};
CK_ULONG outputFormat = CK_OTP_FORMAT_DECIMAL;
CK_ULONG outputLength = 6;
CK_DATE endDate = {...};
CK_BYTE counterValue[8] = {0};
CK_BYTE value[] = {...};
CK_BBOOL true = CK_TRUE;
CK_ATTRIBUTE template[] = {
   {CKA_CLASS, &class, sizeof(class)},
   {CKA_KEY_TYPE, &keyType, sizeof(keyType)},
   {CKA_END_DATE, &endDate, sizeof(endDate)},
   {CKA_TOKEN, &true, sizeof(true)},
   {CKA_SENSITIVE, &true, sizeof(true)},
   {CKA_LABEL, label, sizeof(label)-1},
   {CKA_SIGN, &true, sizeof(true)},
   {CKA_VERIFY, &true, sizeof(true)},
   {CKA_ID, keyId, sizeof(keyId)},
   {CKA_OTP_FORMAT, &outputFormat,
   sizeof(outputFormat)},
   {CKA_OTP_LENGTH, &outputLength,
   sizeof(outputLength)},
   {CKA_OTP_COUNTER, counterValue,
   sizeof(counterValue)},
   {CKA_VALUE, value, sizeof(value)}
};
```

5.4.2 ACTI key generation

The ACTI key generation mechanism, denoted **CKM_ACTI_KEY_GEN**, is a key generation mechanism for the ACTI algorithm.

It does not have a parameter.

The mechanism generates ACTI keys with a particular set of attributes as specified in the template for the key.

The mechanism contributes at least the **CKA_CLASS**, **CKA_KEY_TYPE**, **CKA_VALUE** and **CKA_VALUE_LEN** attributes to the new key. Other attributes supported by the ACTI key type may be specified in the template for the key, or else are assigned default initial values.

For this mechanism, the *ulMinKeySize* and *ulMaxKeySize* fields of the **CK_MECHANISM_INFO** structure specify the supported range of ACTI key sizes, in bytes.

5.4.3 ACTI OTP generation and validation

CKM_ACTI is the mechanism for the retrieval and verification of ACTI OTP values.

The mechanism takes a pointer to a **CK_OTP_PARAMS** structure as a parameter.

When signing or verifying using the **CKM_ACTI** mechanism, *pData* shall be set to NULL_PTR and *ulDataLen* shall be set to 0.

When verifying an OTP value using the **CKM_ACTI** mechanism, *pSignature* shall be set to the OTP value itself, e.g. the value of the **CK_OTP_VALUE** component of a **CK_OTP_PARAMS** structure in the case of an earlier call to **C_Sign**.

A. Manifest constants

Note: A C or C++ source file in a Cryptoki application or library can define all the types, mechanisms, and other constants described here by including the header file otp-pkcs11.h. When including the otp-pkcs11.h header file, it should be preceded by an inclusion of the top-level Cryptoki header file pkcs11.h, and the source file must also specify the preprocessor directives indicated in Section 8 of [1].

A.1 Object classes

```
#define CKO_OTP_KEY                     0x00000008
```

A.2 Key types

```
#define CKK_SECURID                     0x00000022
#define CKK_HOTP                        0x00000023
#define CKK_ACTI                        0x00000024
```

A.3 Mechanisms

```
#define CKM_SECURID_KEY_GEN             0x00000280
#define CKM_SECURID                     0x00000282

#define CKM_HOTP_KEY_GEN                0x00000290
#define CKM_HOTP                        0x00000291

#define CKM_ACTI_KEY_GEN                0x000002A0
#define CKM_ACTI                        0x000002A1
```

A.4 Attributes

```
#define CKA_OTP_FORMAT                   0x00000220
#define CKA_OTP_LENGTH                   0x00000221
#define CKA_OTP_TIME_INTERVAL            0x00000222
#define CKA_OTP_USER_FRIENDLY_MODE       0x00000223
#define CKA_OTP_CHALLENGE_REQUIREMENT    0x00000224
#define CKA_OTP_TIME_REQUIREMENT         0x00000225
#define CKA_OTP_COUNTER_REQUIREMENT      0x00000226
#define CKA_OTP_PIN_REQUIREMENT          0x00000227
#define CKA_OTP_USER_IDENTIFIER          0x0000022A
#define CKA_OTP_SERVICE_IDENTIFIER       0x0000022B
#define CKA_OTP_SERVICE_LOGO             0x0000022C
#define CKA_OTP_SERVICE_LOGO_TYPE        0x0000022D
#define CKA_OTP_COUNTER                  0x0000022E
#define CKA_OTP_TIME                     0x0000022F
```

A.5 Attribute constants

```
#define CK_OTP_FORMAT_DECIMAL           0
```

```
#define CK_OTP_FORMAT_HEXADECIMAL    1
#define CK_OTP_FORMAT_ALPHANUMERIC   2
#define CK_OTP_FORMAT_BINARY         3

#define CK_OTP_PARAM_IGNORED         0
#define CK_OTP_PARAM_OPTIONAL        1
#define CK_OTP_PARAM_MANDATORY       2
```

A.6 Other constants

```
#define CK_OTP_VALUE                 0
#define CK_OTP_PIN                   1
#define CK_OTP_CHALLENGE             2
#define CK_OTP_TIME                  3
#define CK_OTP_COUNTER               4
#define CK_OTP_FLAGS                 5
#define CK_OTP_OUTPUT_LENGTH         6
#define CK_OTP_FORMAT                7

#define CKF_NEXT_OTP                 0x00000001
#define CKF_EXCLUDE_TIME             0x00000002
#define CKF_EXCLUDE_COUNTER          0x00000004
#define CKF_EXCLUDE_CHALLENGE        0x00000008
#define CKF_EXCLUDE_PIN              0x00000010
#define CKF_USER_FRIENDLY_OTP        0x00000020
```

A.7 Notifications

```
#define CKN_OTP_CHANGED              1
```

A.8 Return values

```
#define CKR_NEW_PIN_MODE   0x000001B0
#define CKR_NEXT_OTP       0x000001B1
```

B. Example code

B.1 Disclaimer concerning sample code

For the sake of brevity, sample code presented herein is somewhat incomplete. In particular, initial steps needed to create a session with a cryptographic token are not shown, and the error handling is simplified.

B.2 OTP retrieval

The following sample code snippet illustrates the retrieval of an OTP value from an OTP token using the **C_Sign** function. The sample demonstrates the generality of the approach described herein and does not include any OTP mechanism-specific knowledge.

```
    CK_SESSION_HANDLE hSession;
    CK_OBJECT_HANDLE hKey;
```

```
CK_RV rv;
CK_SLOT_ID slotId;
CK_OBJECT_CLASS class = CKO_OTP_KEY;
CK_ATTRIBUTE template[] = {
  {CKA_CLASS, &class, sizeof(class)} };
CK_UTF8CHAR time[] = {...};
/* UTC time value for OTP, or NULL */
CK_UTF8CHAR pin[] = {...};
/* User PIN, or NULL */
CK_BYTE counter[] = {...};
/* Counter value, or NULL */
CK_BYTE challenge[] = {...};
/* Challenge, or NULL */
CK_MECHANISM_TYPE_PTR allowedMechanisms = NULL_PTR;
CK_MECHANISM_INFO mechanismInfo;
CK_MECHANISM mechanism;
CK_ULONG i, ulOTPLen, ulKeyCount, ulChalReq, ulPINReq,
      ulTimeReq, ulCounterReq;
CK_ATTRIBUTE mechanisms[] = { {CKA_ALLOWED_MECHANISMS,
      NULL_PTR, 0} };
CK_ATTRIBUTE attributes[] = {
  {CKA_OTP_CHALLENGE_REQUIREMENT, &ulChalReq,
      sizeof(ulChalReq)},
  {CKA_OTP_PIN_REQUIREMENT, &ulPINReq,
      sizeof(ulPINReq)},
  {CKA_OTP_COUNTER_REQUIREMENT, &ulCounterReq,
      sizeof(ulCounterReq)},
  {CKA_OTP_TIME_REQUIREMENT, &ulTimeReq,
      sizeof(ulTimeReq)} };

CK_OTP_PARAM param[4];
CK_OTP_PARAMS params;
CK_BYTE *pOTP; /* Storage for OTP result */

do {

  /* N.B.: Minimal error and memory handling in this
     sample code. */

  /* Find first OTP key on the token. */
  if ((rv = C_FindObjectsInit(hSession, template, 1))
      != CKR_OK) {
    break;
  };
  if ((rv = C_FindObjects(hSession, &hKey, 1,
      &ulKeyCount)) != CKR_OK) {
    break;
```

```
};
if (ulKeyCount == 0) {
  /* No OTP key found */
  break;
}
rv = C_FindObjectsFinal(hSession);

/* Find a suitable OTP mechanism. */
if ((rv = C_GetAttributeValue(hSession, hKey,
    mechanisms, 1)) != CKR_OK) {
  break;
};

if ((allowedMechanisms = (CK_MECHANISM_TYPE_PTR)
    malloc(mechanisms[0].ulValueLen)) == 0) {
  break;
};

mechanisms[0].pValue = allowedMechanisms;
if ((rv = C_GetAttributeValue(hSession, hKey,
    mechanisms, 1)) != CKR_OK) {
  break;
};

for (i = 0; i < mechanisms[0].ulValueLen/
    sizeof(CK_MECHANISM_TYPE); ++i) {
  if ((rv = C_GetMechanismInfo(slotId,
    allowedMechanisms[i], &mechanismInfo)) ==
    CKR_OK) {
    if (mechanismInfo.flags & CKF_SIGN) {
      break;
    }
  }
}

if (i == mechanisms[0].ulValueLen) {
  break;
}

mechanism.mechanism = allowedMechanisms[i];
free(allowedMechanisms);

/* Set required mechanism parameters based on
   the key attributes. */
if ((rv = C_GetAttributeValue(hSession, hKey,
     attributes, sizeof(attributes) /
     sizeof(attributes[0]))) != CKR_OK) {
```

```
    break;
  }

  i = 0;
  if (ulPINReq == CK_OTP_PARAM_MANDATORY) {
    /* PIN value needed. */
    param[i].type = CK_OTP_PIN;
    param[i].pValue = pin;
    param[i++].ulValueLen = sizeof(pin) - 1;
  }
  if (ulChalReq == CK_OTP_PARAM_MANDATORY) {
    /* Challenge neded. */
    param[i].type = CK_OTP_CHALLENGE;
    param[i].pValue = challenge;
    param[i++].ulValueLen = sizeof(challenge);
  }
  if (ulTimeReq == CK_OTP_PARAM_MANDATORY) {
    /* Time needed (would not normally be
       the case if token has its own clock). */
    param[i].type = CK_OTP_TIME;
    param[i].pValue = time;
    param[i++].ulValueLen = sizeof(time) -1;
  }
  if (ulCounterReq == CK_OTP_PARAM_MANDATORY) {
    /* Counter value needed (would not normally
       be the case if token has its own counter.*/
    param[i].type = CK_OTP_COUNTER;
    param[i].pValue = counter;
    param[i++].ulValueLen = sizeof(counter);
  }

  params.pParams = param;
  params.ulCount = i;

  mechanism.pParameter = &params;
  mechanism.ulParameterLen = sizeof(params);

  /* Sign to get the OTP value. */
  if ((rv = C_SignInit(hSession, &mechanism, hKey))
    != CKR_OK) {
    break;
  }

  /* Get the buffer length needed for the OTP Value
     and any associated data. */
  if ((rv = C_Sign(hSession, NULL_PTR, 0, NULL_PTR,
      &ulOTPLen)) != CKR_OK) {
        break;
      };

      if ((pOTP = malloc(ulOTPLen)) == NULL_PTR) {
        break;
      };

      /* Get the actual OTP value and any
         associated data. */
      if ((rv = C_Sign(hSession, NULL_PTR, 0, pOTP,
               &ulOTPLen)) != CKR_OK) {
        break;
      }

      /* Traverse the returned pOTP here. The actual
         OTP value is in CK_OTP_VALUE in pOTP. */

    } while (0);
```

B.3 User-friendly mode OTP token

This sample demonstrates an application retrieving a user-friendly OTP value. The code is the same as in B.1 except for the following:

```
/* Add these variable declarations */

  CK_FLAGS flags = CKF_USER_FRIENDLY_OTP;
  CK_BBOOL bUserFriendlyMode;
  CK_ULONG ulFormat;

/* Replace the declaration of the "attributes" and the
   "param" variables with: */

  CK_ATTRIBUTE attributes[] = {
    {CKA_OTP_CHALLENGE_REQUIREMENT, &ulChalReq,
    sizeof(ulChalReq)},
    {CKA_OTP_PIN_REQUIREMENT, &ulPINReq,
    sizeof(ulPINReq)},
    {CKA_OTP_COUNTER_REQUIREMENT, &ulCounterReq,
    sizeof(ulCounterReq)},
    {CKA_OTP_TIME_REQUIREMENT, &ulTimeReq,
    sizeof(ulTimeReq)},
    {CKA_OTP_USER_FRIENDLY_MODE, &bUserFriendlyMode,
    sizeof(bUserFriendlyMode)},
    {CKA_OTP_FORMAT, &ulFormat,
    sizeof(ulFormat)}
  };
```

```
        CK_OTP_PARAM param[5];

    /* Replace the assignment of the "pParam" component
       of the "params" variable with: */

        if (bUserFriendlyMode == CK_TRUE) {
          /* Token supports user-friendly OTPs */
          param[i].type = CK_OTP_FLAGS;
          param[i].pValue = &flags;
          param[i++].ulValueLen = sizeof(CK_FLAGS);
        } else if (ulFormat == CK_OTP_FORMAT_BINARY) {
          /* Some kind of error since a user-friendly
             OTP cannot be returned to an application
             that needs it. */
          break;
        };

        params.pParams = param;

    /* Further processing is as in B.1. */
```

B.4 OTP verification

The following sample code snippet illustrates the verification of an OTP value from an RSA SecurID token, using the **C_Verify** function. The desired UTC time, if a time is specified, is supplied in the CK_OTP_PARAMS structure, as is the user's PIN.

```
    CK_SESSION_HANDLE hSession;
    CK_OBJECT_HANDLE hKey;
    CK_UTF8CHAR time[] = {...};
    /* UTC time value for OTP, or NULL */
    CK_UTF8CHAR pin[] = {...};
    /* User PIN or NULL (if collected by library) */
    CK_OTP_PARAM param[] = {
      {CK_OTP_TIME, time, sizeof(time)-1},
      {CK_OTP_PIN, pin, sizeof(pin)-1}
    };
    CK_OTP_PARAMS params = {param, 2};
    CK_MECHANISM mechanism = {CKM_SECURID, &params,
            sizeof(params)};
    CK_ULONG ulKeyCount;
    CK_RV rv;
    CK_BYTE OTP[] = {...};     /* Supplied OTP value. */
    CK_ULONG ulOTPLen = strlen((CK_CHAR_PTR)OTP);
    CK_OBJECT_CLASS class = CKO_OTP_KEY;
    CK_KEY_TYPE keyType = CKK_SECURID;
```

```
   CK_ATTRIBUTE template[] = {
     {CKA_CLASS, &class, sizeof(class)},
     {CKA_KEY_TYPE, &keyType, sizeof(keyType)},
   };

   /* Find the RSA SecurID key on the token. */
   rv = C_FindObjectsInit(hSession, template, 2);
   if (rv == CKR_OK) {
      rv = C_FindObjects(hSession, &hKey, 1, &ulKeyCount);
      rv = C_FindObjectsFinal(hSession);
   }

   if ((rv != CKR_OK) || (ulKeyCount == 0)) {
      printf(" \nError: unable to find RSA SecurID key on
           token.\n");
      return(rv);
}

   rv = C_VerifyInit(hSession, &mechanism, hKey);
   if (rv == CKR_OK) {
     ulOTPLen = sizeof(OTP);
     rv = C_Verify(hSession, NULL_PTR, 0, OTP, ulOTPLen);
   }

   switch(rv) {
       case CKR_OK:
           printf("\nSupplied OTP value verified.\n");
           break;

       case CKR_SIGNATURE_INVALID:
           printf("\nSupplied OTP value not verified.\n");
           break;

       default:
           printf("\nError:Unable to verify OTP value.\n");
           break;
   }

   return(rv);
```

C. Intellectual property considerations

RSA Security makes no patent claims on the general constructions described in this document, although specific underlying techniques may be covered. The RSA SecurID technology is covered by a number of US patents (and foreign counterparts), in particular US patent nos. 4,856,062, 4,885,778, 5,097,505, 5,168,520, and 5,657,388. Additional patents are pending.

This document and the information contained herein are provided on an "AS IS" basis and RSA SECURITY DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. RSA Security makes no representations regarding intellectual property claims by other parties. Such determination is the responsibility of the user.

D. References

[1] RSA Laboratories, *PKCS #11: Cryptographic Token Interface Standard*. Version 2.20, June 2004. URL: ftp://ftp.rsasecurity.com/pub/pkcs/pkcs-11/v2-20/pkcs-11v2-20.pdf.

[2] Rigney et al, "Remote Authentication Dial In User Service (RADIUS)", IETF RFC2865, June 2000. URL: http://ietf.org/rfc/rfc2865.txt.

[3] Aboba et al, "Extensible Authentication Protocol (EAP)", IETF RFC 3748, June 2004. URL: http://ietf.org/rfc/rfc3748.txt.

E. About OTPS

The *One-Time Password Specifications* are documents produced by RSA Laboratories in cooperation with secure systems developers for the purpose of simplifying integration and management of strong authentication technology into secure applications, and to enhance the user experience of this technology.

Further development of the OTPS series will occur through mailing list discussions and occasional workshops, and suggestions for improvement are welcome. As four our PKCS documents, results may also be submitted to standards forums. For more information, contact:

OTPS Editor
RSA Laboratories
174 Middlesex Turnpike
Bedford, MA 01730 USA
otps-editor@rsasecurity.com
http://www.rsasecurity.com/rsalabs/

EXHIBIT 7

Samsung
Knox

White Paper: An Overview of the Samsung KNOX™ 2.0 Platform

March 2014
Enterprise Mobility Solutions
Samsung Electronics Co., Ltd.

SAMSUNG

Contents

Acronyms 1

Introducing the Samsung KNOX™ 2.0 Platform 2

What's New in the KNOX 2.0 Platform 3

Technology Overview 3

1. Platform Security 4

- Secure Boot and Trusted Boot 4
- Security Enhancements for Android 4
- TrustZone-based Integrity Measurement Architecture 5

2. Application Security 6

- Trust-Zone-based Security Services 6
- KNOX Container 7
- Virtual Private Network Support 9
- SmartCard Framework 10
- Single Sign-On 11

3. Mobile Device Management 11

- Comprehensive Management Policies 11
- Simplified Enrollment 13

4. Certifications 14

- FIPS 140-2 Certification 14
- DISA MOS SRG Compliance 14
- Common Criteria Certification 14

Summary 15

About Samsung Electronics Co., Ltd. 16

Acronyms

AES	Advanced Encryption Standard
BYOD	Bring Your Own Device
CAC	U.S. Common Access Card
COPE	Corporate-Owned Personally Enabled
DAR	Data-at-Rest
DISA	U.S. Defense Information Systems Agency
DIT	Data-in-Transit
DoD	U.S. Department of Defense
FIPS	Federal Information Processing Standard
IPC	Inter Process Communication
MAC	Mandatory Access Control
MDM	Mobile Device Management
NIST	National Institute of Standards and Technology
ODE	On-Device Encryption
PKCS	Public Key Cryptography Standards
ROM	Read-Only Memory
SBU	Sensitive But Unclassified
SE for Android	Security Enhancements for Android
SE Linux	Security-Enhanced Linux
SRG	Security Requirements Guide
SSO	Single Sign-On
STIGs	Security Technical Implementation Guides
TIMA	TrustZone-based Integrity Measurement Architecture
VPN	Virtual Private Network

Introducing the Samsung KNOX™ 2.0 Platform

Samsung KNOX 2.0 is the next-generation of the secured Android platform introduced by Samsung in 2013 as Samsung KNOX. Targeted primarily at mid and high-tier devices, it leverages hardware security capabilities to offer multiple levels of protection for the operating system and applications.

Key features include Trusted Boot, ARM® TrustZone® -based Integrity and Security services, SE for Android enhancements, and the KNOX 2.0 container.

In addition, KNOX 2.0 features a new enterprise enrollment process that vastly improves both the employee and IT administrator experience for enrolling devices into the company's MDM system.

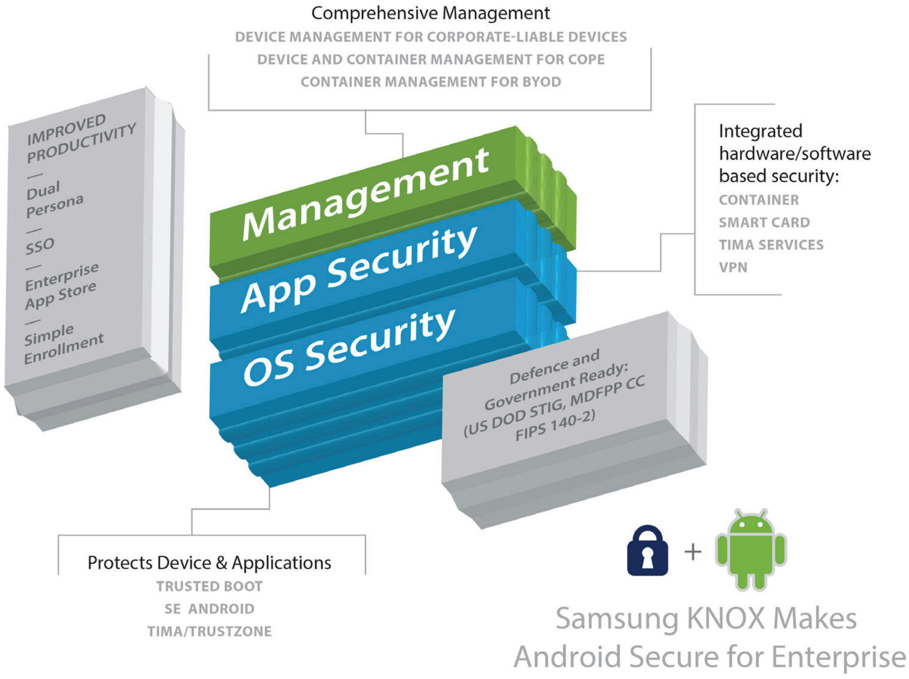

Comprehensive Management
DEVICE MANAGEMENT FOR CORPORATE-LIABLE DEVICES
DEVICE AND CONTAINER MANAGEMENT FOR COPE
CONTAINER MANAGEMENT FOR BYOD
IMPROVED PRODUCTIVITY
—
Dual Persona
—
SSO
—
Enterprise App Store
—
Simple Enrollment
Management
App Security
OS Security
Integrated hardware/software based security:
CONTAINER
SMART CARD
TIMA SERVICES
VPN
Defence and Government Ready: (US DOD STIG, MDFPP CC FIPS 140-2)
Protects Device & Applications
TRUSTED BOOT
SE ANDROID
TIMA/TRUSTZONE
+
Samsung KNOX Makes
Android Secure for Enterprise

Figure 1 – Samsung KNOX 2.0 Platform

The KNOX 2.0 platform offers several new security and management features.

What's New in the KNOX 2.0 Platform

The KNOX 2.0 platform includes a number of new features that address key enterprise needs. In response to requests for additional security features, the platform includes:

- SE for Android protection for third-party containers that enterprise may have already deployed
- TrustZone-based KeyStore to provide hardware-based protection for encryption keys
- TrustZone-based Client Certificate Management for hardware-protected certificate management
- TrustZone-based On-Device Encryption to verify system integrity at boot time before data decryption occurs

The user experience for enterprise enrollment of Android devices has generally lagged behind that of other mobile platforms. The KNOX 2.0 platform now offers a unified enrollment option that MDM vendors can leverage to offer their customers a simple and intuitive experience.

In addition, several features of the original KNOX 1.0 platform have been enhanced to offer additional security features to enterprises. These enhancements include:

- Real-time kernel protection in addition to periodic kernel monitoring
- Major enhancements to the KNOX container that eliminates wrapping, features more management policies, and allows for more flexible data sharing
- A multi-vendor Virtual Private Network (VPN) framework with a variety of third-party clients including SSL VPN are available
- An open SmartCard framework that enables enterprises to choose from an array of SmartCard readers

Technology Overview

This section describes the technical aspects of three key features of Samsung KNOX 2.0 platform:

1. Platform Security
2. Application Security
3. Mobile Device Management

1. Platform Security

Samsung KNOX addresses security using a comprehensive, three-prong strategy:

- Secure Boot and Trusted Boot
- Security Enhancements for Android (SE for Android)
- TrustZone-based Integrity Measurement Architecture (TIMA)

Trusted Boot, SE for Android, and TIMA are the cornerstones of KNOX security.

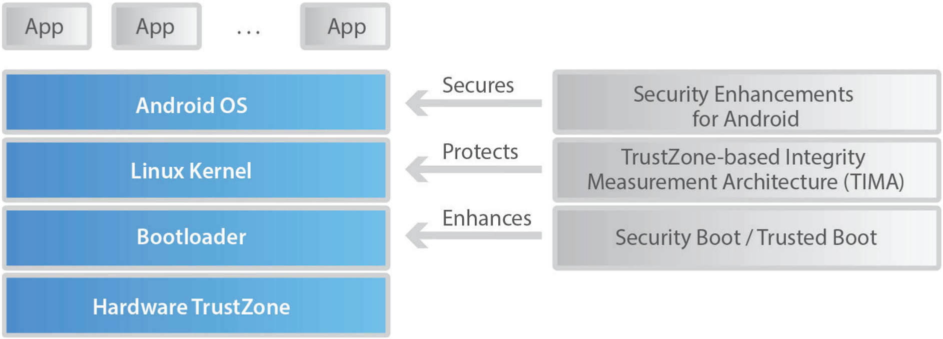

App
App
. . .
App
Android OS
Linux Kernel
Bootloader
Hardware TrustZone
Secures
Protects
Enhances
Security Enhancements for Android
TrustZone-based Integrity Measurement Architecture (TIMA)
Security Boot / Trusted Boot

Figure 2 – Samsung KNOX 2.0 Platform Security Overview

1. Platform Security

- **Secure Boot and Trusted Boot**
- Security Enhancements for Android
- TrustZone-based Integrity Measurement Architecture

The startup process for Android begins with the *primary* bootloader, which is loaded from ROM. This code performs basic system initialization and then loads another bootloader, called a *secondary* bootloader, from the file system into RAM and executes it. Multiple secondary bootloaders may be present, each for a specific task. The boot process is sequential in nature with each secondary bootloader completing its task and executing the next secondary bootloader in the sequence, finally loading the Android bootloader known as *aboot*, which loads the Android operating system.

Secure Boot is a security mechanism that prevents unauthorized bootloaders and operating systems from loading during the startup process. Secure Boot is implemented by each bootloader cryptographically verifying the next bootloader in the sequence using a certificate chain that has its root-of-trust resident in the hardware. The boot process is terminated if verification fails at any step.

Typically, the bootloader verification process is only performed until aboot is loaded, which itself does not verify the Android operating system. This allows users to install and boot customized versions of Android OS kernels. As a result, there is no guarantee for enterprise users that their Android system is enforcing OS-level security protection, such as SE for Android, which is essential for protecting enterprise apps and data.

Samsung KNOX 2.0 implements Trusted Boot to address this limitation of Secure Boot. With Trusted Boot, measurements of the bootloaders are recorded in secure memory during the boot process. At runtime, TrustZone applications use these measurements to make security-critical decisions, such as verifying the release of security keys, container activation, and so on.

Additionally, if the aboot bootloader is unable to verify the Android kernel, a one-time programmable memory area (colloquially called a *fuse*) is written to indicate suspected tampering. Even if the boot code is restored to its original factory state, this evidence of tampering remains. However, the boot process is not halted, and the aboot bootloader continues to boot the Android operating system. This process ensures that normal operation of the device is not affected.

1. Platform Security

- Secure Boot and Trusted Boot
- **Security Enhancements for Android**
- TrustZone-based Integrity Measurement Architecture

Samsung KNOX 2.0 utilizes SE for Android to enforce Mandatory Access Control (MAC) policies to isolate applications and data within the platform. While Google also introduced SE for Android in version 4.4 of the Android platform, Samsung's implementation provides significant enhancements in the level of protection offered to applications and system services. The Google SE for Android policy defines 48 security domains, of which only four domains enforce policies while the others operate in the so-called permissive mode of SELinux. In contrast, KNOX SE for Android Policy defines over 100 security domains that strictly enforce security policies.

The KNOX 2.0 platform includes real-time kernel protection.

The KNOX 2.0 platform introduces a new feature called SE for Android Management Service (SEAMS) that provides controlled access to the SELinux policy engine. SEAMS is used internally by the KNOX 2.0 container, and is also available to third-party vendors to secure their own container solutions. For security considerations, the domains for third-party containers are defined *a priori* by Samsung and activated on-demand when the container application is first invoked. SEAMS also provides enterprises the ability to replace individual SELinux policy files. This feature is governed by a special KNOX license and intended only for very specialized environments.

The system protection offered by SE for Android relies on the assumption of OS kernel integrity. If the kernel itself is compromised (by a perhaps as yet unknown future vulnerability), SE for Android security mechanisms could potentially be disabled and rendered ineffective. Samsung's TrustZone-based Integrity Measurement Architecture (TIMA) was developed to close this vulnerability. TIMA leverages hardware features, specifically TrustZone, to ensure that it cannot be preempted or disabled by malicious software.

TIMA Periodic Kernel Monitoring (PKM)

TIMA PKM performs continuous periodic monitoring of the kernel to detect if legitimate kernel code and data have been modified by malicious software. In addition, TIMA also monitors key SE for Android data structures in OS kernel memory to prevent malicious attacks from corrupting them and potentially disabling SE for Android.

TIMA Real-time Kernel Protection (RKP)

TIMA RKP performs ongoing, strategically-placed real-time monitoring of the operating system from within TrustZone to prevent tampering of the kernel. RKP intercepts critical events happening inside the kernel, which are inspected in TrustZone. If an event is determined to have impact on the integrity of the OS kernel, RKP either stops the event, or logs an attestation verdict that tampering is suspected, which is sent to the MDM. This protects against malicious modifications and injections to kernel code, including those that coerce the kernel into corrupting its own data.

Remote Attestation

Attestation has many similarities to Trusted Boot and essentially uses the same fundamental data sources and procedures. The primary difference is that Attestation can be requested on-demand by the enterprise's Mobile Device Management (MDM) system.

When requested, Attestation reads the previously-stored measurement information and the fuse value (see Trusted Boot above), then combines the data in a proprietary way to produce an Attestation verdict. This verdict, essentially a coarse indication that tampering is suspected, is returned to the requesting MDM. The cryptographic signature is based on the device's unique Attestation Certificate, and embedded in the device during the manufacturing process. This process ensures that the Attestation verdict cannot be altered during transfer.

Any further action is determined by the enterprise's MDM security policy. The security policy might choose to detach from the device, erase the contents of the secure application container, ask for the location of the device, or any of many other possible security recovery procedures.

KNOX 2.0 leverages TrustZone to offer enhanced security to applications.

2. Application Security

In addition to securing the platform, Samsung KNOX 2.0 provides solutions to address the security needs of individual applications:

- TrustZone-based Security Services
- KNOX 2.0 container
- Virtual Private Network Support
- SmartCard Framework
- Single Sign-On

2. Application Security
- TrustZone-based Security Services
- KNOX container
- Virtual Private Network Support
- SmartCard Framework
- Single Sign-On

TrustZone-based Client Certificate Management (CCM)

TrustZone-based CCM enables storage and retrieval of digital certificates, as well as other operations using those certificates such as encryption, decryption, signing, verification, and so on, in a manner similar to the functions of a SmartCard. The certificates and associated keys are encrypted with a device-unique hardware key that can only be decrypted from code running within TrustZone.

TrustZone-based CCM also provides the ability to generate a Certificate Signing Request (CSR) and the associated public/private key pairs in order to obtain a digital certificate. A default certificate is provided for applications that do not require their own certificate.

Programming interfaces for certificate storage and management are provided in the KNOX Premium SDK. Application developers are provided with industry standard PKCS #11 APIs for signing and encryption, and therefore interact with the CCM as if it were a virtual SmartCard. Both types of operations are permitted only if Trusted Boot can guarantee system integrity.

TrustZone-based KeyStore

The KeyStore provides applications with services for generating and maintaining cryptographic keys. The keys are further encrypted with a device-unique hardware key that can only be decrypted by the hardware from within TrustZone. All cryptographic operations are performed only within TrustZone, and are disabled if the system is compromised, as determined by Trusted Boot.

Application developers should continue to use the familiar Android KeyStore APIs and specify that the KeyStore is used to provide the service.

TrustZone-based On-Device Encryption

The KNOX 2.0 platform further strengthens the full-device encryption capability offered by the Android platform. In addition to successful password authentication, the system integrity as determined by Trusted Boot is also verified before the data is decrypted.

This feature is available only if the enterprise IT administrator activates encryption via the MDM. TrustZone-based On-Device Encryption also enables enterprises to ensure that all device data is protected in the unlikely event that the operating system is compromised.

The KNOX 2.0 container runs unmodified Android applications.

2. Application Security

- TrustZone-based Security Services
- KNOX container
- Virtual Private Network Support
- SmartCard Framework
- Single Sign-On

The Samsung KNOX 2.0 container provides a separate Android environment within the mobile device, complete with its own home screen, launcher, applications, and widgets.

Applications and data inside the container are isolated from applications outside the container, that is, applications outside the container cannot use Android inter-process communication or data-sharing methods with applications inside the container. For example, photos taken with the camera inside the container are not viewable in the Gallery outside the container. The same restriction applies to copying and pasting. Note that the contacts and calendar apps represent an exception, since container contacts and the calendar can be made visible inside the KNOX container and in the personal work space. The end user can choose whether to share contacts and calendar notes between the container and personal space, however, IT policy ultimately controls this option.

The enterprise can manage the container like any other IT asset using an MDM solution; this container management process is called Mobile Container Management (MCM). Samsung KNOX 2.0 supports many of the leading MDM solutions on the market. MCM is affected by setting policies in the same fashion as traditional MDM policies. The Samsung KNOX 2.0 container includes a rich set of policies for authentication, data security, VPN, e-mail, application blacklisting, whitelisting, and so on.

The KNOX 2.0 platform features major enhancements to the KNOX container in the KNOX 1.0 platform. The most significant enhancement is elimination of application wrapping. This is achieved by leveraging technology introduced by Google in Android 4.2 to support multiple users on tablet devices. This enhancement enables enterprises to easily deploy custom applications without requiring Samsung to wrap the application. It also further reduces the barrier to entry for independent software developers wishing to develop and deploy applications for the KNOX 2.0 container.

The new container also adds a two-factor authentication process. The user can create a finger print to access the container and select either a PIN, password, or pattern as a second process to follow the finger print.

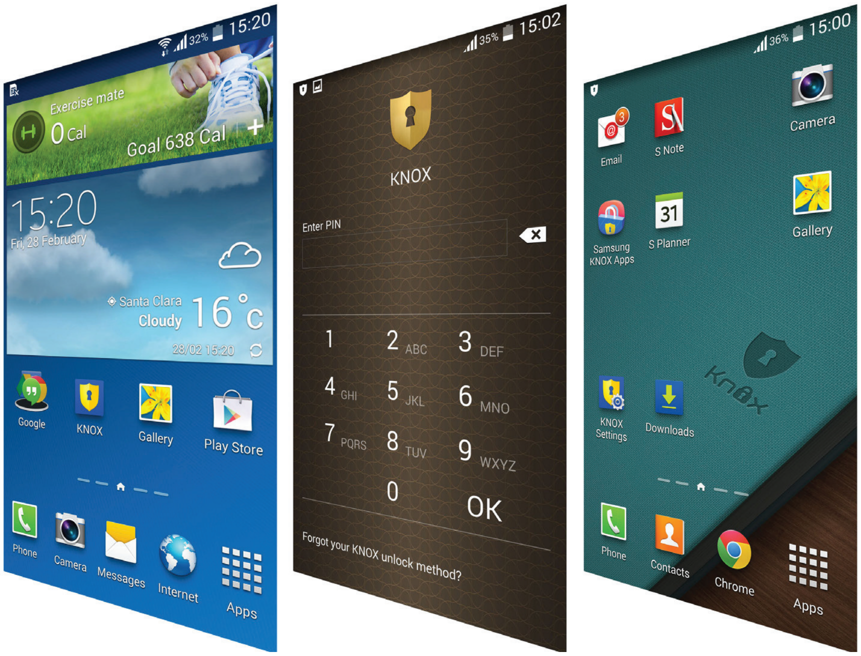
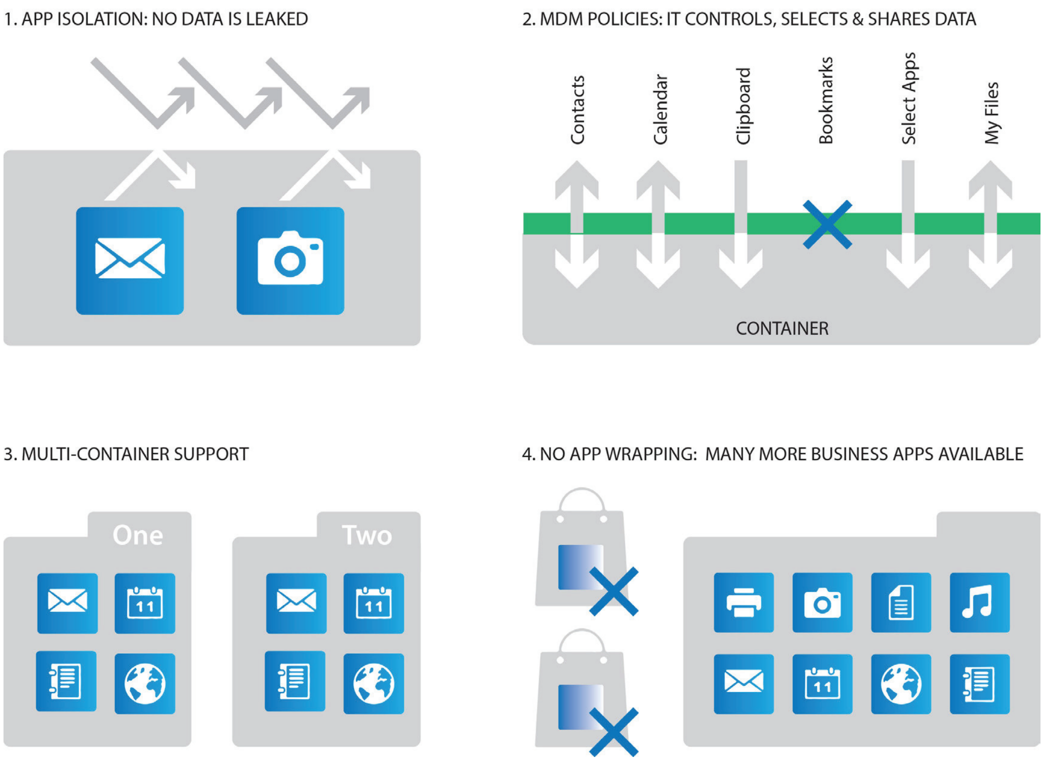


Figure 3 – Samsung KNOX 2.0 Container

The KNOX 2.0 container allows enterprises to balance security and user productivity.

The KNOX 2.0 platform also introduces support for multiple containers, thus meeting the needs of professionals that use their own devices for corporate use (BYOD) and have multiple employers, such as doctors or consultants.

The new KNOX 2.0 container also allows enterprise IT administrators to control the flow of information between the container and the rest of the device. This feature enables enterprises to strike the right balance between security and user productivity. Users can also control the device's data sharing capability based on their personal preferences, according to the limits specified by enterprise IT administrators.



1. APP ISOLATION: NO DATA IS LEAKED

2. MDM POLICIES: IT CONTROLS, SELECTS & SHARES DATA

3. MULTI-CONTAINER SUPPORT

4. NO APP WRAPPING: MANY MORE BUSINESS APPS AVAILABLE

Figure 4 – Samsung KNOX 2.0 Container

KNOX offers stronger VPN support for both IPSec and SSL VPNs.

2. Application Security
- TrustZone-based Security Services
- KNOX container
- Virtual Private Network Support
- SmartCard Framework
- Single Sign-On

The KNOX 2.0 platform offers additional comprehensive support for enterprise Virtual Private Networks (VPN). This support enables businesses to offer their employees an optimized, secure path to corporate resources from their BYOD or Corporate-Owned Personally Enabled (COPE) devices.

The KNOX 1.0 platform offered broad support for the IPSec protocol suite including features such as:

- Internet Key Exchange (IKE and IKEv2)
- Triple DES (56/168-bit), AES (128/256-bit) encryption
- Split tunneling mode
- Suite B Cryptography

However, a large number of enterprises have deployed Secure Socket Link (SSL) VPNs to enable remote access to their workforce as they do not require the full connectivity to the enterprise network, but rather a small set of resources such as web-based applications and file shares.

The KNOX 2.0 platform adds support for leading SSL VPN vendors. As SSL implementations are proprietary, KNOX 2.0 features a new generic VPN framework which enables third-party SSL vendors to provide their clients as plug-ins into the VPN framework. Enterprise IT administrators use KNOX MDM policies to download and configure a specific SSL client.

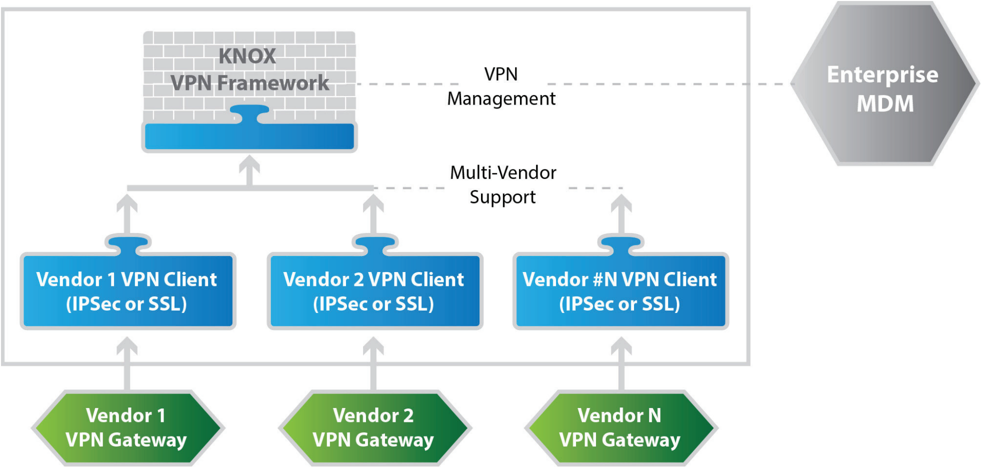


Figure 5 – Multi-Vendor Support in KNOX

The per-application VPN feature in the KNOX 1.0 platform has been extended to support SSL VPNs. This feature enables the enterprise to automatically enforce the use of VPN only on a specific set of applications. For example, the enterprise IT administrator can configure an employee's device to enforce VPN for only business applications. This feature ensures that the data from the user's personal applications do not use the VPN and overload the company's intranet. At the same time, user privacy is preserved because personal data does not use the enterprise network.

The KNOX 2.0 platform gives comprehensive support across SmartCard readers.

The per-app VPN feature can also be applied to the KNOX 2.0 container either for all or a subset of the applications in the container.

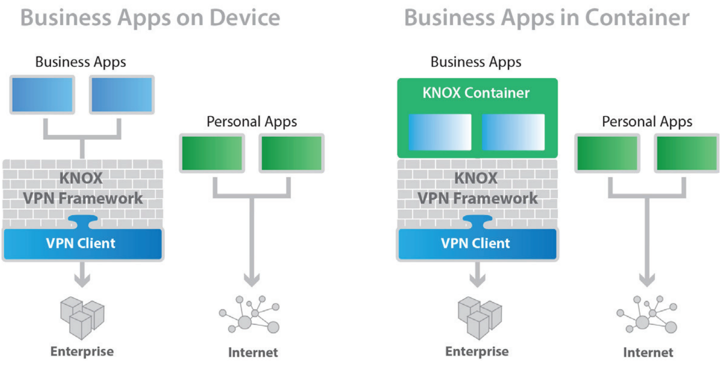


Figure 6 – Per Application VPN in KNOX 2.0

2. Application Security

- TrustZone-based Security Services
- KNOX container
- Virtual Private Network Support
- SmartCard Framework
- Single Sign-On

The United States Department of Defense (US DoD) has mandated the use of Public Key Infrastructure (PKI) certificates for employees to digitally sign documents, encrypt and decrypt e-mail messages, and establish secure online network connections. These certificates are typically stored on a SmartCard called the Common Access Card (CAC).

The Samsung KNOX 2.0 platform provides applications access to the hardware certificates on the CAC via standards-based Public Key Cryptography Standards (PKCS) APIs. This access process enables the use of the CAC card by the browser, e-mail application, and VPN client, as well as other custom government applications.

Other enterprises have growing interest to use SmartCards for the same purpose, especially those that require high levels of security and information protection.

The KNOX 2.0 platform provides improved SmartCard compatibility via a new software framework that allows third-party SmartCard and reader providers to install their solutions into the framework.

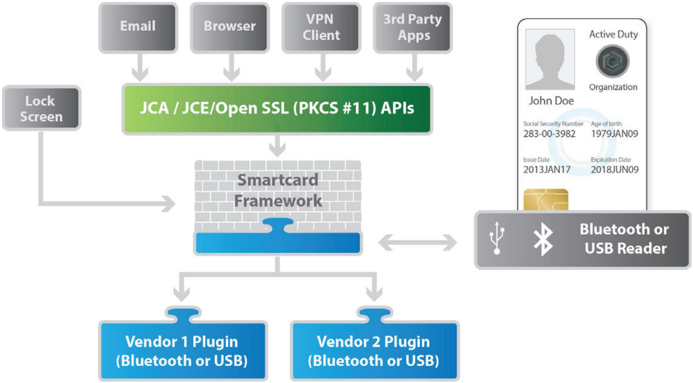


Figure 7 – Samsung KNOX 2.0 Support for SmartCards

Single Sign-On (SSO) increases security and reduces IT costs.

2. Application Security

- TrustZone-based Security Services
- KNOX container
- Virtual Private Network Support
- SmartCard Framework
- Single Sign-On

Single Sign-On (SSO) is access control of several related, but independent software systems. The user logs in once and has access to all systems without being prompted to log in again for each application. For example, SSO allows access to the container and apps within the container with one password.

SSO shares centralized authentication servers that all other applications and systems use for authentication purposes. It combines this with techniques to ensure that users do not have to actively enter their credentials more than once.

Advantages of using SSO include:

- Reduces the number of user names and password combinations a user must remember
- Reduces time spent re-entering passwords for the same user
- Reduces IT costs with less help desk calls about passwords
- Increases security because tokens and certificates are transmitted over the internet for authentication as opposed to plain text passwords.

3. Mobile Device Management

Enrolling mobile devices into the enterprise network and remote management of these devices are key aspects of an enterprise mobility strategy. The KNOX 2.0 platform addresses both of these requirements:

- Comprehensive management with over 530 policies
- Simplified enrollment for a faster and intuitive user experience

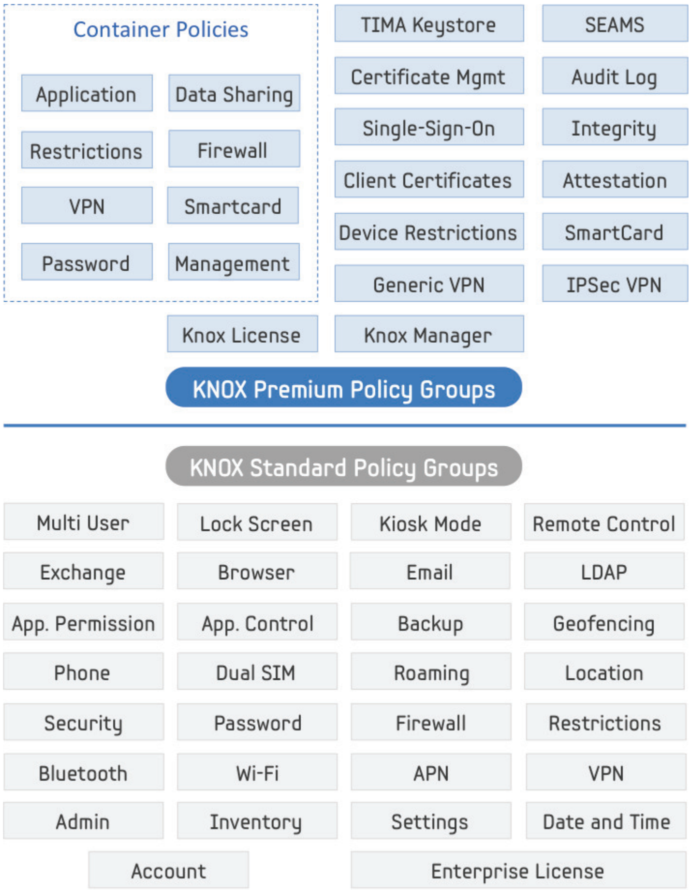
3. Mobile Device Management

- Comprehensive Management Policies
- Simplified Enrollment

The KNOX 2.0 platform offers significant enhancements to the management policies offered in the KNOX 1.0 platform. The various policy groups are classified into two major categories: Standard and Premium.

The Standard Policy suite represents continuous enhancements Samsung developed over Google Android management capability since 2009. The SDK for these policy APIs is available to MDM vendors and other interested ISVs free of charge. Further, no runtime license fee is associated with these APIs.

KNOX 2.0 offers comprehensive management capabilities for the enterprise IT administrator.



Container Policies

Application | Data Sharing
Restrictions | Firewall
VPN | Smartcard
Password | Management

TIMA Keystore | SEAMS
Certificate Mgmt | Audit Log
Single-Sign-On | Integrity
Client Certificates | Attestation
Device Restrictions | SmartCard
Generic VPN | IPSec VPN

Knox License | Knox Manager

KNOX Premium Policy Groups

KNOX Standard Policy Groups

Multi User	Lock Screen	Kiosk Mode	Remote Control
Exchange	Browser	Email	LDAP
App. Permission	App. Control	Backup	Geofencing
Phone	Dual SIM	Roaming	Location
Security	Password	Firewall	Restrictions
Bluetooth	Wi-Fi	APN	VPN
Admin	Inventory	Settings	Date and Time
Account		Enterprise License	

Figure 8 – Samsung KNOX 2.0 Management Policies

The KNOX 2.0 Premium Policy suite is the collection of policy groups offering advanced capabilities such as management and control of the KNOX container, security features such as the KeyStore and Client Certificate Manager, Per-application VPN, and so on. The SDK for these policy APIs is also available at no charge, however, enterprises using these features are required to purchase a KNOX License that is verified on the device at runtime.

Samsung KNOX has simplified the enterprise enrollment process.

3. Mobile Device Management

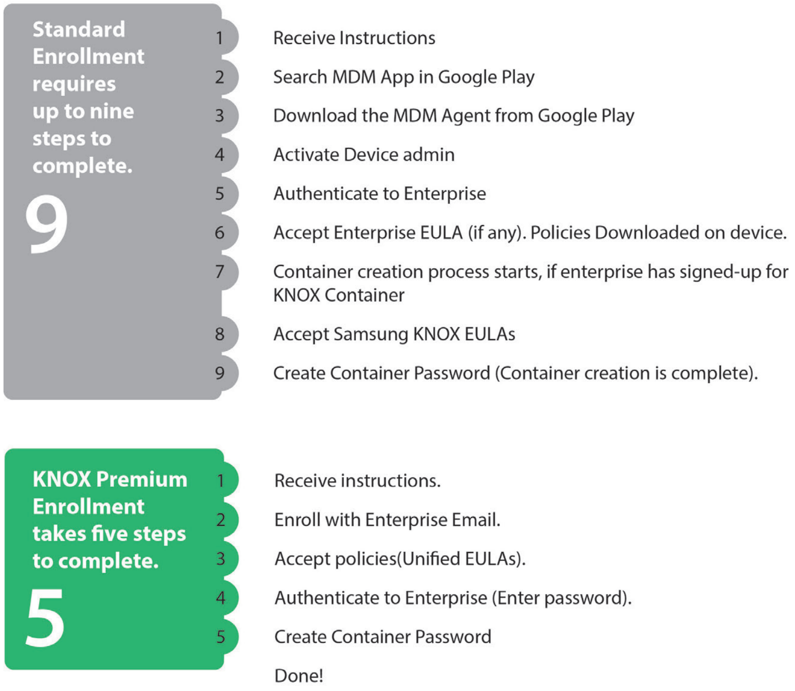
- Comprehensive Management Policies
- Simplified Enrollment

Enrolling an Android device into a company's MDM system typically begins with a user downloading the agent application from the Google Play store, then configuring it for authentication. Enterprises are facing increasing help desk calls as more and more users are activating mobile devices for work. When presented with prompts, privacy policies, and license agreements, users might experience difficulties during the process, resulting in a poor overall experience.

The KNOX 2.0 platform provides a simplified enrollment solution that is simple and intuitive and eliminates many steps and human error.

The simplified enrollment process provides the employee with an enrollment link sent by e-mail, text message, or through the company's internal or external website. Once the link is clicked, users are prompted to enter their corporate e-mail address. This action triggers the display of all required privacy policies and agreements. After accepting the terms, users enter a corporate account password for authentication from the enterprise. Any agent application required is automatically downloaded and installed.

MDM vendors can take advantage of this feature to simplify the onboarding process for enterprise users, significantly improve the user experience, and reduce support costs.



Standard Enrollment requires up to nine steps to complete. 9

1 Receive Instructions
2 Search MDM App in Google Play
3 Download the MDM Agent from Google Play
4 Activate Device admin
5 Authenticate to Enterprise
6 Accept Enterprise EULA (if any). Policies Downloaded on device.
7 Container creation process starts, if enterprise has signed-up for KNOX Container
8 Accept Samsung KNOX EULAs
9 Create Container Password (Container creation is complete).

KNOX Premium Enrollment takes five steps to complete. 5

1 Receive instructions.
2 Enroll with Enterprise Email.
3 Accept policies(Unified EULAs).
4 Authenticate to Enterprise (Enter password).
5 Create Container Password
Done!

Figure 9 – KNOX 2.0 Container Simplified Enrollment

Samsung KNOX is ready for deployment in high security environments.

Certifications

Issued by the National Institute of Standards and Technology (NIST), the Federal Information Processing Standard (FIPS) is a US security standard that helps ensure companies that collect, store, transfer, share, and disseminate sensitive but unclassified (SBU) information and controlled unclassified information (CUI) can make informed purchasing decisions when choosing devices to use in their workplace.

Samsung KNOX 2.0 meets the requirements for FIPS 140-2 Level 1 certification for both data-at-rest (DAR) and data-in-transit (DIT).

The Defense Information Systems Agency (DISA) is an agency within the US DoD that publishes Security Requirements Guides (SRGs) as processes to improve the security of DoD information systems. SRGs guide the development of Security Technical Implementation Guides (STIGs) which document specific product policies and requirements as well as best practices for configuration. In 2012, DISA published the Mobile Operating System SRG to specify the security requirements that commercially available mobile devices should meet in order to be deployed within the DoD.

On May 2, 2013 DISA approved the STIG for Samsung KNOX drafted for the Mobile Operating System SRG.

The Common Criteria for Information Technology Security Evaluation, commonly referred to as Common Criteria, is an internally-recognized standard for defining security objectives of information technology products and for evaluating vendor compliance with these objectives. A number of Governments use Common Criteria as the basis for their own certification schemes.

Select Galaxy devices with KNOX embedded received Common Criteria (CC) certification on February 27, 2014. The current CC certification targets the new Mobile Device Fundamentals Protection Profile (MDFPP) of the National Information Assurance Partnership (NIAP), published in October 2013, which addresses the security requirements of mobile devices for use in enterprise.

Summary

The Samsung KNOX platform addressed several CIO concerns about security and management of Android devices:

- Trusted Boot, TIMA, and SE for Android protect the operating system and platform services from malware attacks and hacking
- The KNOX 2.0 container provides enhanced security to enterprise applications by preventing data leakage
- The per-application VPN features enables enterprises to enforce secure VPN connectivity only for corporate apps.
- The rich set of MDM policies enables enterprise IT administrators to comprehensively manage the device

The new KNOX 2.0 platform further raises the bar on security, manageability, and ease-of-use with several new features and enhancements:

- Real-time kernel protection against malicious kernel attacks
- Hardware-backed storage for cryptography keys and client certificates
- Remote attestation capability that allows enterprises to verify the authenticity and integrity of KNOX devices during and after enrollment
- The KNOX 2.0 container runs unmodified Android applications and eliminates the need for application wrapping
- Enterprise-controllable data sharing between personal space and enterprise container
- A multi-vendor VPN framework that allows a variety of third-party clients including SSL VPN
- An open SmartCard framework that allows enterprises to choose from an array of SmartCard readers

These and numerous other enhancements make the new KNOX 2.0 platform the most secure and enterprise-ready Android platform, whether devices are employee-owned (BYOD) or corporate-issued (COPE).

About Samsung Electronics Co., Ltd.

Samsung Electronics Co., Ltd. is a global leader in technology, opening new possibilities for people everywhere. Through relentless innovation and discovery, we are transforming the worlds of televisions, smartphones, personal computers, printers, cameras, home appliances, LTE systems, medical devices, semiconductors and LED solutions. We employ 236,000 people across 79 countries with annual sales exceeding KRW 201 trillion. To discover more, please visit www.samsung.com

For more information about Samsung KNOX,
Visit www.samsung.com/knox

Samsung Electronics Co., Ltd.
416, Maetan 3-dong, Yeongtong-gu
Suwon-si, Gyeonggi-do 443-772, Korea

EXHIBIT 8

SAMSUNG

Knox

White Paper: An Overview of the Samsung Knox™ Platform

August 2016
Samsung Research America
Samsung Electronics Co., Ltd.

SAMSUNG

Contents

Samsung Knox Platform 4
Technology Overview 5
Platform Security 5
Hardware Root of Trust 5
Secure Boot and Trusted Boot 6
Security Enhancements for Android 6
TrustZone-based Integrity Measurement Architecture 7
TrustZone-based Security Services 9
Application Security 10
Knox Workspace 10
Knox Enabled App (KEA) 14
Android for Work on a Samsung device 15
Virtual Private Network Support 15
SmartCard Framework 18
Single Sign-On 18
Mobile Device Management 19
Comprehensive Management Policies 20
Active Directory Integration 20
Knox Mobile Enrollment 21
Enterprise Billing 21
Certifications 23
About Samsung Electronics Co., Ltd. 25

Acronyms

AES	Advanced Encryption Standard
AOSP	Android Open Source Project
BYOD	Bring Your Own Device
CAC	U.S. Common Access Card
CESG	Communications and Electronic Security Group
COPE	Corporate-Owned Personally Enabled
DAR	Data-at-Rest
DISA	U.S. Defense Information Systems Agency
DIT	Data-in-Transit
DoD	U.S. Department of Defense
FIPS	Federal Information Processing Standard
IPC	Inter Process Communication
KEA	Knox Enabled App
MAC	Mandatory Access Control
MDM	Mobile Device Management
NIST	National Institute of Standards and Technology
ODE	On-Device Encryption
OS	Operating System
PKCS	Public Key Cryptography Standards
RAM	Random-Access Memory
ROM	Read-Only Memory
SBU	Sensitive But Unclassified
SE for Android	Security Enhancements for Android
SE Linux	Security-Enhanced Linux
SRG	Security Requirements Guide
SSO	Single Sign-On
STIGs	Security Technical Implementation Guides
TIMA	TrustZone-based Integrity Measurement Architecture
VPN	Virtual Private Network

Samsung Knox™ Platform

Knox is Samsung's defense-grade mobile security platform built into our newest devices. Just turn on the device, and you're protected.

Knox provides strong guarantees for the protection of enterprise data by building a hardware-rooted *trusted environment*. A trusted environment ensures that enterprise-critical operations, such as decryption of enterprise data, can only occur when core system components are proven to not be compromised. For many pieces of device software, such as the kernel and TrustZone apps, this is done by checking the cryptographic signature of each piece of software. A trusted environment is hardware-rooted if both the cryptographic keys and code used to compute these signatures are tied back to unmodifiable values stored in hardware.

Key features of Knox include Secure Boot, Trusted Boot, ARM® TrustZone® -based Integrity Measurement Architecture (TIMA), Security Enhancements for Android (SE for Android), and TrustZone-based Security Services.

The Knox Workspace container is designed to separate, isolate, encrypt, and protect work data from attackers. This enterprise-ready solution provides management tools and utilities to meet security needs of enterprises large and small.

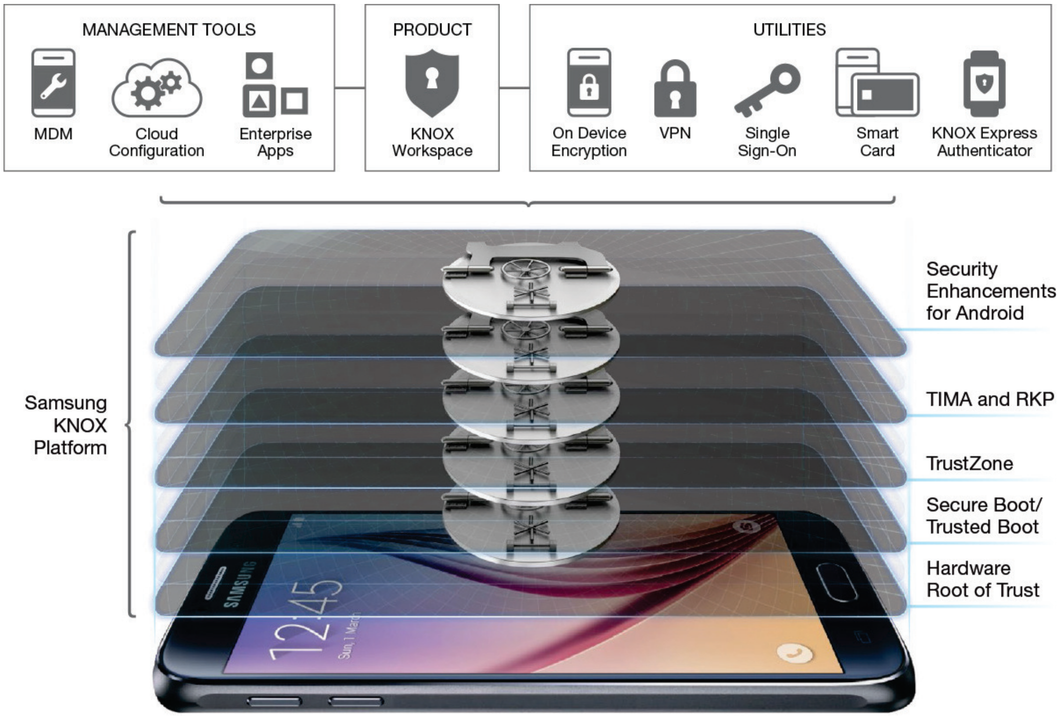


Figure 1 – Samsung Knox Platform, Workspace, Management Tools and Utilities

Technology Overview

This section describes the technical aspects of three key pillars of Samsung Knox platform:

1. Platform Security
2. Application Security
3. Mobile Device Management

Platform Security

Samsung Knox addresses security using a comprehensive, hardware-rooted trusted environment:

- Hardware Root of Trust
- Secure Boot and Trusted Boot
- Security Enhancements for Android (SE for Android)
- TrustZone-based Integrity Measurement Architecture (TIMA)
- TrustZone-based Security Services

Hardware Root of Trust

Three hardware components are the foundation of Samsung Knox's trusted environment.

The Device Root Key (DRK) is a device-unique asymmetric key that is signed by Samsung through an X.509 certificate. This certificate attests that the DRK was produced by Samsung. The DRK is injected in the device at manufacture time in the Samsung factory, and is only accessible by specially privileged software modules within the TrustZone Secure World. Because the DRK is device-unique, it can be used to identify a device. For example, a certificate included with TIMA attestation data is signed by DRK (more precisely, through a key attested by the DRK), which proves that the attestation data originated from the TrustZone Secure World on a Samsung device. Knox also uses device-unique hardware keys and keys derived from the hardware keys, which are only accessible in the TrustZone Secure World. Such keys can be used to tie data to a device. For example, Knox Workspace data is encrypted by such a key, and it cannot be decrypted on any other devices.

The Samsung Secure Boot key is used to sign Samsung-approved executables of boot components. The public part of the Samsung Secure Boot key is stored in hardware at manufacture time in the Samsung factory. The Secure Boot process uses this public key to verify whether each boot component it loads is approved.

Rollback prevention fuses are hardware fuses that encode the minimum acceptable version of Samsung-approved executables. These fuses are set at manufacture time in the Samsung factory. Because old images may contain known vulnerabilities that can be exploited, this feature prevents approved-but-old versions of boot executables from being loaded.

Secure Boot and Trusted Boot

The startup process for Android begins with the primary bootloader, which is loaded from Read-only Memory (ROM). This code performs basic system initialization and then loads another bootloader, called a secondary bootloader, from the file system into Random-Access Memory (RAM) and executes it. Multiple secondary bootloaders may be present, each for a specific task. The boot process is sequential in nature with each secondary bootloader completing its task and executing the next secondary bootloader in the sequence, finally loading the Android bootloader known as *aboot*, which loads the Android operating system.

Secure Boot is a security mechanism that prevents unauthorized bootloaders and operating systems from **loading** during the startup process. Secure Boot is implemented by each bootloader cryptographically verifying the signature of the next bootloader in the sequence using a certificate chain that has its root-of-trust resident in hardware. The boot process is terminated if verification fails at any step.

Secure Boot is effective in preventing unauthorized bootloaders (and sometimes the kernel when it is also applied to the kernel binary). However, Secure Boot is unable to distinguish between different versions of authorized binaries, for example, a bootloader with a known vulnerability versus a later patched version, since both versions have valid signatures. In addition, when some carriers decide to allow custom kernels to run on their devices, Secure Boot is not effective in preventing non-Samsung kernels from running on these devices. This exposes an attack surface that poses a potential threat to enterprise applications and data.

Samsung Knox implements Trusted Boot (in addition to Secure Boot) to address this limitation. With Trusted Boot, measurements of the bootloaders are recorded in secure memory during the boot process. At runtime, TrustZone applications use these measurements to make security-critical decisions, such as verifying the release of cryptographic keys from the TIMA KeyStore, container activation, and so on.

Additionally, if the *aboot* bootloader is unable to verify the Android kernel, a one-time programmable memory area (colloquially called a fuse) is written to indicate suspected tampering. Even if the boot code is restored to its original factory state, this evidence of tampering remains. However, the boot process is not halted, and the *aboot* bootloader continues to boot the Android operating system. This process ensures that normal operation of the device is not affected.

Security Enhancements for Android

Samsung Knox introduced Security Enhancements for Android (SE for Android) in 2012 to enforce Mandatory Access Control (MAC) policies. These enhancements protect applications and data by strictly defining what each process is allowed to do, and which data it can access. Samsung's innovative collaborations with the authors of SELinux resulted in the gold standard for Android security. In version 4.4 of Android Open Source Project (AOSP), Google introduced a subset of the SE for Android enhancements Samsung pioneered (i.e., the SELinux portion). Samsung continues to lead Google, and all others, in continuing to implement new extensions of SE for Android. Our improvements allow us to protect areas of the Android framework to which access was previously unrestricted. Our policy protects software created by Samsung, AOSP, and other third-party partners. The increased enforcement granularity from our AOSP enhancements, and Samsung's industry-leading granular access policies that define over 200 unique security domains, are designed together to enforce the tightest restrictions with the lowest rates of over- or under-privileging.

Samsung also built an innovative global policy validation system that can detect when prohibited actions are attempted. This gives us unique visibility into how our devices are used and can alert us to new threats. This system can be used to refine our policy and very accurately grant only the minimum permissions needed.

The Knox platform now includes the SE for Android Management Service (SEAMS) that provides Application Programming Interface (API)-level control of the security policy engine. SEAMS is primarily used internally by the Knox Workspace container, but is also available to third-party vendors to secure their own container solutions. The SEAMS APIs allow software permissions to be tailored for each organization. Leveraging our controls to define and protect security containers allows customers to dynamically isolate applications and data. Our containers use new SE for Android Multi-Level Security (MLS) protections designed to offer far more protection than any other existing Android isolation mechanism.

Our enhancements to Android, along with our robust policy, security tools, data collection, and policy management show that Samsung Knox devices are designed to provide the best protections of any mobile device manufacturer. These protections form a foundation for protecting users from malicious or accidental security breaches.

TrustZone-based Integrity Measurement Architecture

The system protection offered by SE for Android relies on the assumption of Operating System (OS) kernel integrity. If the kernel itself is compromised (by a perhaps as yet unknown future vulnerability), SE for Android security mechanisms could potentially be disabled and rendered ineffective. Samsung's TrustZone-based Integrity Measurement Architecture (TIMA) was developed to close this vulnerability. TIMA leverages hardware features, specifically TrustZone, to ensure that it cannot be preempted or disabled by malicious software.

TIMA Periodic Kernel Measurement (PKM)

TIMA PKM performs continuous periodic monitoring of the kernel to detect if legitimate kernel code and data have been modified by malicious software. In addition, TIMA also monitors key SE for Android data structures in OS kernel memory to detect malicious attacks that corrupt them and potentially disable SE for Android.

Real-time Kernel Protection (RKP)

RKP performs ongoing, strategically-placed real-time monitoring of the operating system to prevent tampering of the kernel. RKP intercepts critical kernel events, which are then inspected in TrustZone. If an event is determined to have impact on the integrity of the OS kernel, RKP either stops the event, or logs an alert that tampering is suspected. This alert information is included in remote attestation results sent to the MDM for IT admins to determine any further actions required by the enterprises security policies. This protects against malicious modifications and injections to kernel code, including those that coerce the kernel into corrupting its own data. RKP checks are performed in an isolated environment that is inaccessible to the kernel, so potential kernel exploitations cannot be extended to compromise RKP. Depending on the device model, this isolated environment can be in the TrustZone Secure World or the hypervisor extensions.

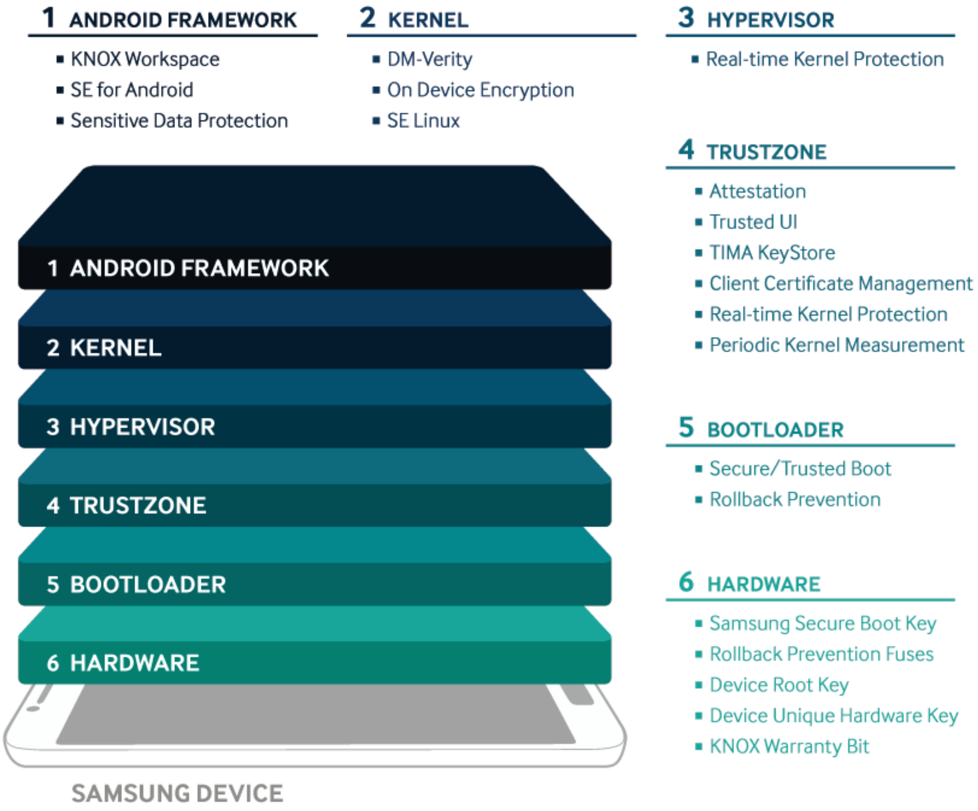


Figure 2 – Samsung Knox Platform Security Overview

Remote Attestation

Remote Attestation (sometimes simply called attestation) is based on Trusted Boot and used to verify the integrity of the platform. Remote attestation can be requested on-demand by the enterprise's Mobile Device Management (MDM) system, typically before creating the Knox Workspace.

When requested, attestation reads the Trusted Boot collected measurement data and returns them to the attestation requestor. To simplify the handling in MDM servers, the attestation agent on the device produces a verdict indicating the overall status of attestation. It compares these measurements to the factory values inside the TrustZone Secure World. Trusted Boot measurement data includes a hardware fuse value that indicates if the device booted into an unauthorized kernel in the past. Trusted Boot measurement data, along with the SE for Android enforcement setting, forms the basis of the produced attestation verdict. This verdict, essentially a coarse indication that tampering is suspected, is returned to the requesting MDM. In addition to the verdict, the attestation data includes all the trusted boot measurements, RKP and PKM logs that can indicate the presence of malicious software in the device, and other device information that can be used to bind the attestation result to the device.

The remote server requesting attestation provides a random nonce to prevent replay attacks. The nonce, the attestation verdict, and the rest of the attestation data are returned to the server, signed with the attestation certificate. The attestation certificate is signed by the Device Root Key (DRK), a device-unique asymmetric key that is signed by a Samsung root key through an X.509 certificate. This chain of certificates ensures that the attestation verdict cannot be altered in transit.

Depending on the attestation verdict and the data, any further action is determined by the enterprise's MDM security policy. The security policy might choose to detach from the device, erase the contents of the secure application container, ask for the location of the device, or any of many other possible security recovery procedures.

TrustZone-based Security Services

TrustZone-based Client Certificate Management (CCM)

TIMA CCM is a TrustZone-based security service also built on the basis of Trusted Boot. A key feature of TIMA CCM is that if the Trusted Boot measurements do not match the authorized values, or if the Knox warranty bit is voided, the entire TIMA CCM functions shut down, ensuring the protection of enterprise data in case of device compromise. TIMA CCM enables storage and retrieval of digital certificates, as well as other operations using those certificates such as encryption, decryption, signing, verification, and so on, in a manner similar to the functions of a SmartCard. The CCM TrustZone code provides a PKCS #11 interface to the Android OS, effectively emulating a smart card interface on a mobile device. The certificates and associated keys are encrypted with a device-unique hardware key that can only be decrypted from code running within TrustZone.

TIMA CCM provides the ability to generate a Certificate Signing Request (CSR) and the associated public/private key pairs in order to obtain a digital certificate. A default certificate is provided for applications that do not require their own certificate. TIMA CCM supports the standard Android Key Chain API, and apps can use CCM by calling APIs that configure Android to use an alternate Key Chain Provider.

All cryptographic components used by CCM are FIPS-140 2 compliant to meet US government requirements for Mobile Device Fundamentals Protection Profile (MDFPP).

TrustZone-based KeyStore

Similar to TIMA CCM, TIMA KeyStore is a TrustZone-based security service also built on the basis of Trusted Boot. The KeyStore provides applications with services for generating and maintaining cryptographic keys. The keys are further encrypted with a device-unique hardware key that can only be decrypted by the hardware from within TrustZone. All cryptographic operations are performed only within TrustZone and are disabled if the system is compromised as determined by Trusted Boot.

TIMA KeyStore supports the Android Key Store API. Application developers can continue to use the familiar Android KeyStore APIs and specify that the TIMA KeyStore is used to provide the service.

TrustZone-based On-Device Encryption

The Knox platform further strengthens the full-device encryption capability offered by the Android platform. In addition to successful password authentication, the system integrity as determined by Trusted Boot is also verified before the data is decrypted. This feature is available only if the enterprise IT admin activates encryption via the MDM. This ensures that all device data is protected in the unlikely event that the operating system is compromised.

Application Security

In addition to securing the platform, Samsung Knox provides solutions to address the security needs of individual applications:

- Knox Workspace
- Virtual Private Network Support
- SmartCard Framework
- Single Sign-On

Knox Workspace

Samsung Knox Workspace is a defense-grade dual persona container product designed to separate, isolate, encrypt, and protect enterprise data from attackers. This work/play environment ensures work data and personal data are separated and that only the work container is managed by the enterprise. Personal information such as photos and messages are not managed or controlled by the IT department. Once activated, the Knox Workspace product is tightly integrated into the Knox platform.

Workspace provides this separate environment within the mobile device, complete with its own home screen, launcher, applications, and widgets.

Applications and data inside Workspace are isolated from applications outside Workspace, that is, applications outside Workspace cannot use Android inter-process communication or data-sharing methods with applications inside Workspace. For example, photos taken with the camera inside Workspace are not viewable in the Gallery outside Workspace. The same restriction applies to copying and pasting. When allowed by IT policy, some application data such as contacts and calendar data can be shared across the Workspace boundary. The end user can choose whether to share contacts and calendar notes between Workspace and personal space; however, IT policy ultimately controls this option. The enterprise can manage Workspace like any other IT asset using an MDM solution. This container management process is called Mobile Container Management (MCM). Samsung Knox supports many of the leading MDM solutions on the market. MCM is affected by setting policies in the same fashion as traditional MDM policies. Samsung Knox Workspace includes a rich set of policies for authentication, data security, VPN, e-mail, application blacklisting, whitelisting, and so on.

The Knox 2.X platform features the elimination of application wrapping, which was used by Knox 1.0 and many other competing solutions. This is achieved by leveraging technology introduced by Google in Android 4.2 to support multiple users on devices. It reduces the barrier to entry for independent software developers wishing to develop and deploy applications for Knox Workspace.

At the time of container creation, IT admins can choose the UI style of the container (folder or launcher style), and can also prevent end users from changing the style.

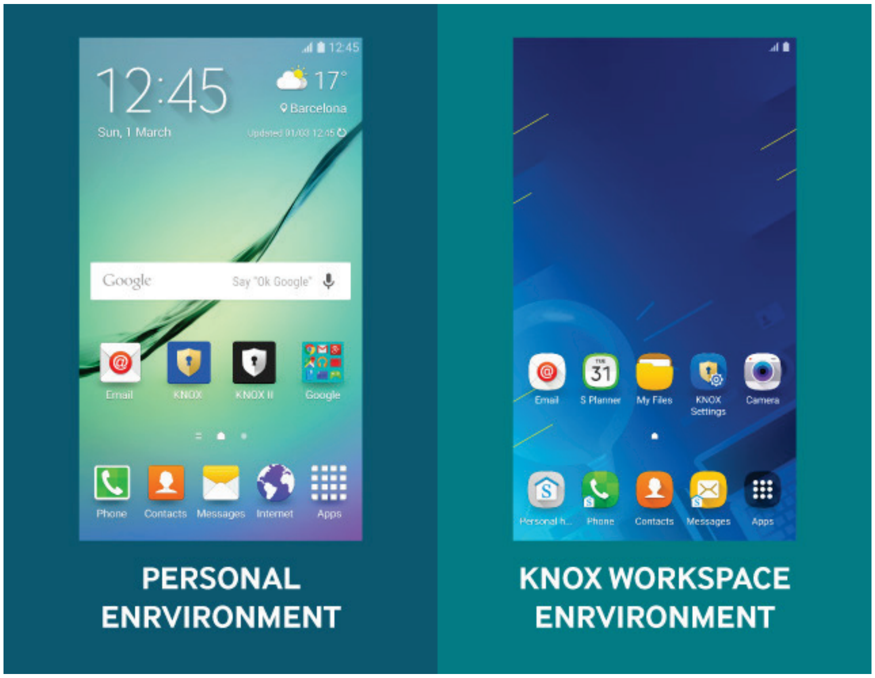


Figure 3 – Samsung Knox Personal Environment and Knox Workspace Environment

My Knox

Samsung My Knox is a free security solution that provides greater separation between enterprise and personal data. In My Knox, you can create a container that only authorized personnel can access. All files and data are encrypted within the container. My Knox is a virtual Android environment within the mobile device complete with its own home screen, launcher, apps, and widgets.

My Knox is not managed by an IT admin or an MDM, but separates work and the personal side of the phone. You can back up data stored on your device to the cloud and restore it to your device when needed.

Secure folder

Secure folder is for consumers to store and access their private apps and data such as photos and email. One installed, the folder contains Contacts, Gallery, Camera, My files, Memo, and a browser. Apps can be added using Google Play.

Knox Workspace can also be configured for container-only mode. In this mode, the entire device experience is restricted to the Workspace. This mode is suitable for industries such as health care, finance, and others who provide devices for employees that seek to restrict access to business applications.

Workspace also has a two-factor authentication process. The user can configure Workspace to accept a fingerprint or iris scan as the primary authentication factor for the container with a PIN, password or pattern as a second factor. The iris scan biometric authentication method is available on the Samsung Galaxy S7.

The Knox platform also supports two containers, thus meeting the needs of professionals that use their own devices for corporate use Bring Your Own Device (BYOD) and have multiple employers, such as doctors or consultants.

IT admins can also enable Bluetooth® and Near Field Communication (NFC) inside Workspace. NFC enables a device to act as a SmartCard-based credential for use cases such as physical access and access to IT accounts. Bluetooth can be used to communicate with connected devices, and supports Bluetooth profiles that enable use cases beyond music and calls inside the Knox Workspace. Examples include printing, file sharing, and external card readers. External SD cards can also be enabled with security restrictions.

Apps inside Workspace can also connect with USB accessories such as a USB printer. For security purposes, IT admins must explicitly allow USB between container apps and external storage. The MDM default for mass storage is set to OFF, and is controlled by enterprise IT admin policy.

For Samsung Note users, S-Pen Air Command is also supported inside Workspace for writing memos, adding app shortcuts (personal apps only), screen capture, and writing notes on a screen capture (depending on IT policy).

Knox caller ID for incoming calls when in Personal mode can also be configured by IT admins to display caller ID information derived from personal contacts and Knox Workspace contacts.

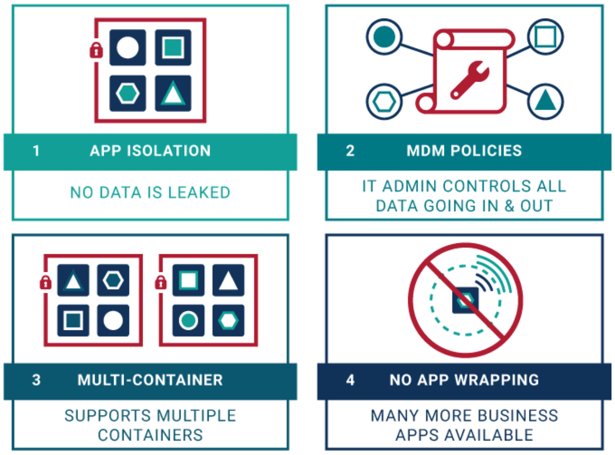

1 APP ISOLATION
NO DATA IS LEAKED
2 MDM POLICIES
IT ADMIN CONTROLS ALL DATA GOING IN & OUT
3 MULTI-CONTAINER
SUPPORTS MULTIPLE CONTAINERS
4 NO APP WRAPPING
MANY MORE BUSINESS APPS AVAILABLE

Figure 4 – Samsung Knox Workspace

Google Play for Work

IT admins can install Google Play for Work inside Knox Workspace for app management to silently install and uninstall apps and blacklist or whitelist apps. Enterprise employees can download apps in Knox Workspace that are approved by IT admins.

Google Voice for apps inside the Knox container allows users to use voice recognition for input in addition to the touchscreen keyboard.

Sensitive Data Protection and Knox Chamber

Knox defines two classes of data – protected and sensitive. All data written by apps in the secure Workspace is protected. Protected data is encrypted on disk when the device is powered off. In addition, the decryption key for protected data is tied to the device hardware. This makes protected data recoverable only on the same device. Furthermore, access controls are used to prevent applications outside the Knox Workspace from attempting to access protected data.

Even stronger protection is applied to sensitive data. Sensitive data remains encrypted as long as the Workspace is locked, even if the device is powered on. When a user unlocks Knox Workspace using their password, Sensitive Data Protection (SDP) allows sensitive data to be decrypted. When the user re-locks the Workspace, SDP keys are cleared. The SDP data decryption key is tied to both device hardware and to the user input. Therefore, the data is recoverable only on the same device and with user input.

SDP can be used in one of two ways. First, all emails received are considered sensitive, and are immediately protected by SDP encryption. Emails received when the Workspace is locked, are immediately encrypted, and can only be decrypted the next time Workspace is unlocked.

The second way to use SDP is through the Knox Chamber. The Chamber is a designated directory on the file system and a user-accessible folder inside Workspace. Any data placed into the Chamber is automatically marked as sensitive and protected by SDP.

Third-party app data can also be encrypted when a device is locked and decrypted when a device is unlocked to prevent data leakage if a device is lost, stolen, or re-used. Keys required for data decryption when unlocking a device are based on the user password.

Knox Quick Access

On Samsung Galaxy S6 devices, based on proximity of a registered and connected Gear device, Knox Quick Access extends the unlock period of the Knox Workspace, thereby reducing the frequency with which the end user must enter password credentials.

Shared device

Many enterprises such as hospitals, banks, and airlines use shared devices for employees. Knox supports the use of shared devices so IT admins can manage device and security policies, and install apps with an MDM. Each employee can login separately with an Active Directory ID and password, which is also integrated with SSO. For security and privacy, all user data is deleted when each employee logs out of the shared device.

Knox Active Protection (KAP)

End users can activate or deactivate Knox Active Protection (KAP) via the Smart Manager app on devices not managed by an MDM. KAP uses both Real-time Kernel Protection (RKP) and DM Verity, a feature that provides integrity checking for system code and data. On MDM-managed devices, KAP is always enabled.

Knox Enabled App (KEA)

Knox Enabled App is a per-app invisible container designed for application developers and vendors to provide security services to Samsung device users. KEA allows service providers to deploy their applications and make maximum use of the Samsung Knox platform security without the need for Mobile Device Management (MDM). Since KEA is an invisible, unmanaged container, the user experience is the same as the original version of the application.Knox platform security extended to KEA provides end users data protection by encrypting app data. If a device is compromised, lost, or stolen, app data cannot be unencrypted.

The KEA workspace is implemented based on Knox Workspace and customized according to use case requirements. Knox Workspace is created and managed by an MDM, and suitable for the enterprise environment. For individual app vendors and developers, creating, managing and configuring the KEA workspace presents challenges without an MDM. However, with KEA, the device automatically creates and manages the KEA workspace when the KEA app is installed.

To operate as a KEA app, additional information (metadata) is required. When a KEA app is installed in KEA-capable devices, the device detects the metadata and authenticates the app through a Knox License Manager (KLM) Server. After authentication is completed, the KEA workspace is created, and the app is installed inside the workspace, including configuration of the SE for Android Management Service (SEAMS) container.

If the KEA app is installed in devices not capable of using KEA, including non-Samsung devices, the KEA metadata is ignored, and the app works as regular Android app, which eliminates the need for a separate version of the app.

KEA Service Flow

1 App developer registers app package name and public key hash on Samsung Enterprise Alliance Program (SEAP) website

2 App developer modifies app according to guidelines

3 App developer uploads app to app store

4 Customers download and install app

5 Samsung verifies license and app

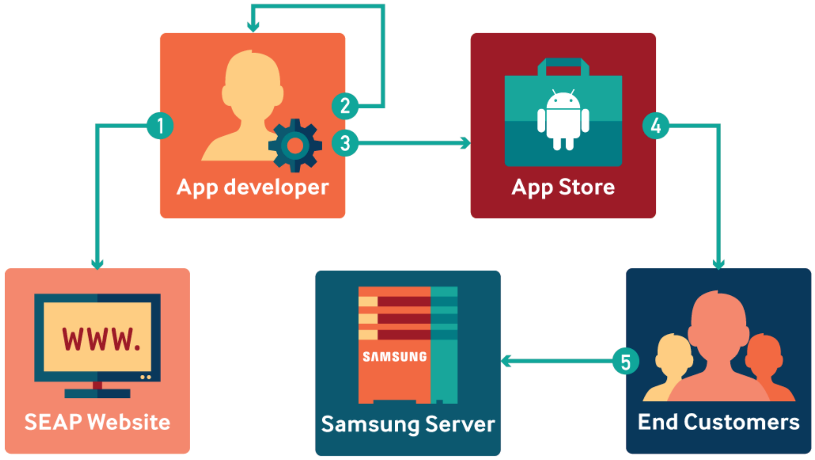


Figure 5 – Service flow of Knox Enabled Apps

Android for Work on a Samsung device

Android for Work Managed Profiles on a Samsung device benefit from key Knox security modules that protect the device and sensitive work data at all times. Knox enables Android for Work protection with the following Knox features:

- RKP actively prevents kernel code modification
- PKM periodically checks kernel code integrity
- DM-Verity verifies the integrity of applications and data stored on the critical system partition
- Trusted Boot measures each software component during boot-time and securely stores the cryptographic hash of the next component in TrustZone memory before loading it.
- Sensitive Data Protection APIs are available for apps in Managed Profiles. The native email app enables SDP once it's installed inside Managed Profiles.
- TIMA and CCM provides the TrustZone-based KeyStore as the default for storing certificates such as VPN and email app certificates.
- Access to Managed Profiles depends on the integrity of the device. If the integrity check fails at the time of creating Android for Work, it is not allowed. If an integrity check fails after Android for Work is installed, the device is not allowed to boot.

Android for Work on a Samsung device does not require a Knox license activation fee. Knox security enhancements for existing Android for Work Managed Profiles are updated seamlessly with Over-the-Air (OTA) updates.

Virtual Private Network Support

The Knox platform offers additional comprehensive support for enterprise Virtual Private Networks (VPN). This support enables businesses to offer their employees an optimized, secure path to corporate resources from their BYOD or Corporate-Owned Personally Enabled (COPE) devices.

Knox offers the following VPN features for IPsec and SSL:

- Per-app connections
- On-demand connections
- Always-on connections
- Device-wide connections
- VPN chaining (nested connections)
- Blocking routes to prevent data leakage if a mandatory VPN connection drops
- Pushing VPN profiles to multiple managed devices
- Traffic usage tracking
- HTTP Proxy

Knox supports the ability to configure VPN connections to enforce redirection of web traffic through an HTTP proxy server, allowing enterprises greater visibility into network traffic and device usage patterns of employees. The Knox VPN framework supports VPN configurations using a static proxy server IP and port, and web proxy authentication.

The Knox platform offers broad feature support for the IPSec protocol suite including:

- Internet Key Exchange (IKE and IKEv2)
- IPsec IETF RFCs – IKEv1
- IKEv1 – Main and aggressive IKE exchange modes with pre-shared key, certificates, Hybrid RSA, and EAP-MD5 authentications
- I IKEv2 with PSK and certificate-based authentication
- IKEv2 – Pre-shared key, certificates, EAP-MD5 EAP-MSCHAPv2 authentication methods, and mobile extensions
- Triple DES (56/168-bit), AES (128/256-bit) encryption with MD5 or SHA
- IKEv1 Suite B Cryptography supported with PSK and ECDS signature-based authentications
- IKEv2 Suite B Cryptography supported with ECDSA signatures

Because a large number of enterprises have deployed Secure Socket Link (SSL) VPNs , the Knox platform provides support for leading SSL VPN vendors. As SSL implementations are proprietary, Knox features a generic VPN framework which enables third-party SSL vendors to provide their clients as plug-ins. Enterprise IT admins use Knox MDM policies to install and configure a specific SSL VPN client.

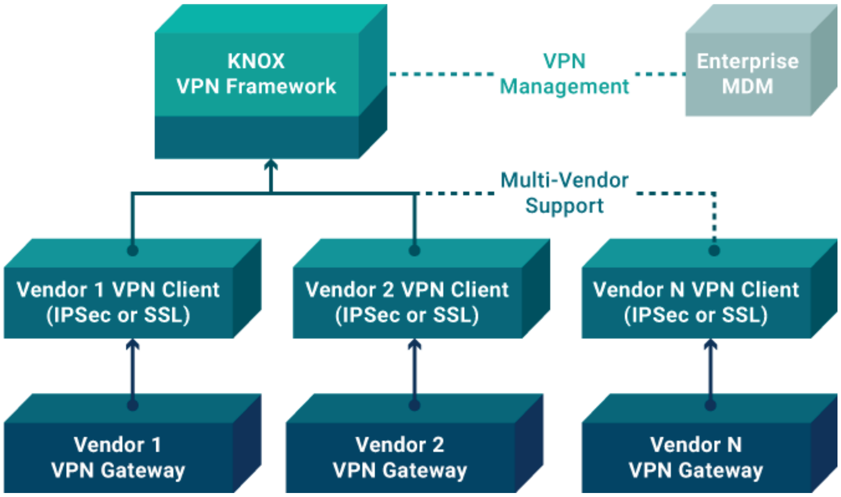


Figure 6 – Multi-Vendor Support in Knox

The per-application VPN feature in the Knox Workspace container enables the enterprise to automatically enforce the use of VPN only on a specific set of applications. For example, an IT admin can configure an employee's device to enforce VPN for only business applications. Such a configuration ensures that the data from the user's personal applications do not use the VPN and overload the company's intranet. At the same time, user privacy is preserved because personal data does not enter the enterprise network.

The per-app VPN feature can also be applied to the Knox Workspace container for all or a subset of the applications in the container.

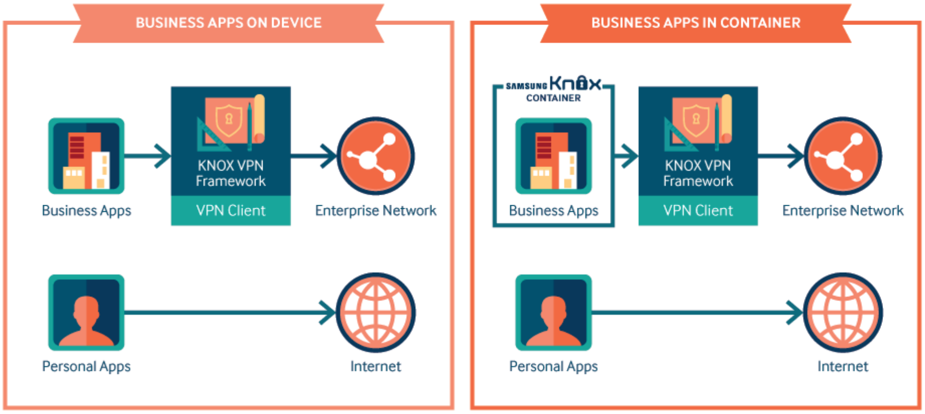


Figure 7 – Per Application VPN in Knox

SmartCard Framework

The United States Department of Defense (US DoD) has mandated the use of Public Key Infrastructure (PKI) certificates for employees to digitally sign documents, encrypt and decrypt e-mail messages, and establish secure online network connections. These certificates are typically stored on a SmartCard called the Common Access Card (CAC).

The Samsung Knox platform provides applications access to the hardware certificates on the CAC via standards-based Public Key Cryptography Standards (PKCS) APIs. This access process enables the use of the CAC card by the browser, e-mail application, and VPN client, as well as other custom government applications.

Other enterprises have growing interest to use SmartCards for the same purpose, especially those that require high levels of security and information protection.

The Knox platform provides improved SmartCard compatibility via a software framework that allows third-party SmartCard and reader providers to install their solutions into the framework.

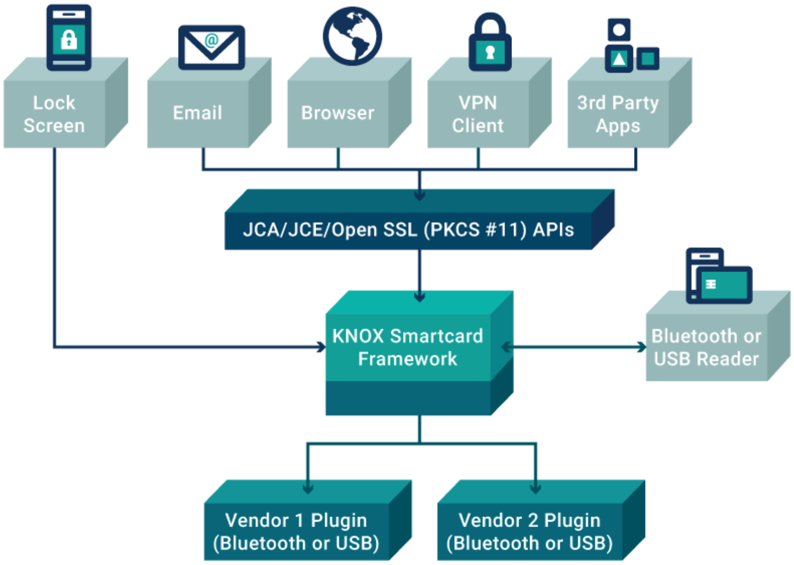

Lock Screen
Email
Browser
VPN Client
3rd Party Apps
JCA/JCE/Open SSL (PKCS #11) APIs
KNOX Smartcard Framework
Bluetooth or USB Reader
Vendor 1 Plugin (Bluetooth or USB)
Vendor 2 Plugin (Bluetooth or USB)

Figure 8 – Samsung Knox support for Smartcards

Single Sign-On

Single Sign-On (SSO) is a feature that provides common access control to several related, but independent software systems. The user logs in once and has access to all systems without being prompted to log in again. For example, SSO allows access to the Workspace container (and participating apps that require credentials within the container) with one password.

SSO shares centralized authentication servers that all other applications and systems use for authentication purposes. It combines this with techniques to ensure that users do not have to actively enter their credentials more than once. SSO reduces the number of user names and passwords a user must remember, and reduces IT costs with fewer help desk calls about login credentials.

Knox Identity and Access Management (IAM) provides a comprehensive and flexible SSO solution to support enterprise applications on Samsung mobile devices. This framework was created to reduce the complexity for enterprise applications to support SSO on mobile devices. There are many Identity Providers with different SSO solutions and with various support protocols such as SAML, OAuth, OpenID, etc. They each distribute their Software Development Kits (SDKs) to mobile app developers, however, developers must customize multiple versions of their apps to support different SSO solutions.

The Knox generic SSO framework is a bridge between the Identity Providers and software developers that allows a single version of an app to work with any SSO solution. The Knox SSO solution provides a unified API for SSO token retrieval and management, called getToken. Samsung partners with leading Identity Partners including Microsoft (Azure Active Directory), CA Technologies, and Centrify. Identity Providers plug their Android Application Package (APK) authenticators into the Knox generic SSO framework and each authenticator works as a proxy to process SSO authentication requests and responses, thereby eliminating the need for developers to create multiple versions of their apps.

Mobile Device Management

Enrolling mobile devices into the enterprise network and remote management of these devices are key aspects of an enterprise mobility strategy. Key device management features of the Knox platform include:

- Comprehensive management with over 1500 MDM APIs
- Active Directory integration
- Knox Mobile Enrollment for a faster and intuitive user experience, including bulk enrollment to assist IT admins to quickly enroll many employees at once
- Enterprise Billing to separate work and personal data costs

Comprehensive Management Policies

The various policy groups are classified into two major categories: Standard and Premium. The Standard Policy suite represents continuous enhancements Samsung developed over Google Android management capability since 2009. The SDK for these policy APIs is available to MDM vendors and other interested ISVs free of charge. Further, no runtime license fee is associated with these APIs.

The Knox Premium Policy suite is the collection of policy groups offering advanced capabilities such as management and control of the Knox Workspace, security features such as the Trusted Boot-based TIMA KeyStore and Client Certificate Manager, Per-application VPN, and so on. The SDK for these policy APIs is also available at no charge; however, enterprises using these features are required to purchase a Knox License that is verified on the device at runtime.

The Knox Audit Log meets MDFPP 2.0 audit requirements. IT admins can select a set of events to audit and periodically push logs to the server.

Some of the events include:

- Administrative actions such as creating containers, password setting policies for devices and containers, app installations and removal
- Certificate failure and key generation
- Adding and removing accounts
- Attempts to exchange files over Wi-Fi

Active Directory integration

Knox provides an option for the IT admin to choose an Active Directory password as the unlock method for Knox containers. This has two important benefits. First, it allows IT admins to use a one-password management policy for desktop and mobile devices. Second, the end user only needs to remember one password to access all services offered by the employer, thereby reducing employee password fatigue and improving productivity.

At the heart of this feature is the proven industry-standard Kerberos protocol. Active Directory is the most widely-deployed enterprise grade directory service that has built-in support for Kerberos. Knox provides a set of Workspace creation parameters to configure Workspace to use the Active Directory password as the unlock method. Additionally, IT admins can also configure Single Sign-on for services inside Workspace, along with the unlock method.

Active Directory passwords can be changed by the user on the mobile device from the settings menu inside the Knox Workspace container. When SSO is configured, the password change does not require entering the password a second time.

Knox Mobile Enrollment

Enrolling an Android device into a company's MDM system typically begins with a user downloading the agent application from the Google Play store, then configuring it for authentication. Enterprises are facing increasing help desk calls as more and more users are activating mobile devices for work. When presented with prompts, privacy policies, and license agreements, users might experience difficulties during the process, resulting in a poor overall experience.

The Knox platform provides a simplified enrollment solution for supported MDMs that is streamlined and intuitive and eliminates many steps and human error.

The enrollment process happens via either self-discovery using an email domain, or employees are provided with an enrollment link sent by e-mail, text message, or through the company's internal or external website. Once the link is clicked, users are prompted to enter their corporate e-mail address. This action triggers the display of all required privacy policies and agreements. After accepting the terms, users enter a corporate account password for authentication from the enterprise. Any agent application required is automatically downloaded and installed.

Samsung Knox Mobile Enrollment allows IT admins to enroll hundreds or thousands of employees at the same time. Samsung provides a web tool and an application to scan package bar codes (the device IMEI). This enrollment method is targeted for devices purchased for COPE enterprises and for supported carriers and resellers.

Another option for IT admins includes using a master device to automatically enroll devices using NFC. The master device is configured by downloading an app from Playstore. Each device is enrolled to an MDM profile selected by the IT admin.

MDM vendors can take advantage of this feature to simplify the onboarding process for enterprise users, significantly improve the user experience, and reduce support costs.

Knox Mobile Enrollment supports multiple MDM configurations per account. With complex device environments, and multiple MDM profiles or configurations, Knox Mobile Enrollment gives IT admins the ability to prepare hundreds of devices and get them connected to the right MDM with ease. End users only need to turn on the device and connect to the network. Knox Mobile Enrollment takes care of activation without users needing to do a thing.

Enterprise Billing

Enterprise Billing provides enterprises a mechanism to separate enterprise data usage from personal data usage. This enables enterprises to compensate their employees for costs generated because of work, particularly in BYOD cases, or to only pay for work-related data in COPE cases.

The Knox platform supports Enterprise Billing from Knox version 2.2 or above, and requires MDM support.

Enterprises configure two Access Point Name (APN) gateways. One APN is for data associated with enterprise-approved apps, and a different APN is for all other personal data. Enterprises must first register with a network operator's enterprise billing service. Once a new APN is provisioned for business use, Knox Workspace can be enabled for that dedicated APN. IT admins can also select individual apps inside or outside Workspace to use data over the enterprise APN.

Enterprise billing configured with a dedicated APN:

- Supports dual-APN Enterprise Billing for carriers using IPv6 networks.
- Separates data usage over the mobile internet for 2G/3G/4G connections
- Routes all data traffic from Knox Workspace over the enterprise APN
- Provides the capability to select individual apps inside or outside Knox Workspace to use data over the enterprise APN

The enterprise APN can also be configured to allow or not allow roaming. When roaming is enabled, personal data is routed through the default APN, and enterprise data is routed through a dedicated enterprise APN. By default, roaming over the enterprise APN is disabled. When a user is roaming in a single Packet Data Protocol (PDP) network, all enterprise apps are automatically routed to the personal APN for work continuity.

If enterprise apps use a VPN connection to the network, the VPN profile can be configured to route data through the enterprise APN.

Dual SIM devices can also be enabled for Knox Enterprise Billing. The primary, or first SIM slot, is automatically selected to configure an APN and activate Enterprise Billing on the device.

To avoid personal use of a SIM card, IT admins can lock the SIM card with a unique PIN combination. This ensures that the SIM can only be used for enterprise billing on the authorized device. In addition, dedicated enterprise APNs are restricted, and APN settings are not visible or editable on the device.

Users can check personal and enterprise data usage on a Knox device in the Settings menu. To view data usage, employees can go to Settings > Data Usage > Mobile Tab (personal) or Enterprise Tab (work).

Certifications

FIPS 140-2 Certification

Issued by the National Institute of Standards and Technology (NIST), the Federal Information Processing Standard (FIPS) is a US security standard that helps ensure companies that collect, store, transfer, share, and disseminate sensitive but unclassified (SBU) information and controlled unclassified information (CUI) can make informed purchasing decisions when choosing devices to use in their workplace.

Samsung Knox meets the requirements for FIPS 140-2 Level 1 certification for both data-at-rest (DAR) and data-in-transit (DIT).

DISA Approved STIG

The Defense Information Systems Agency (DISA) is an agency within the US DoD that publishes Security Technical Implementation Guides (STIGs) which document security policies, requirements, and implementation details for compliance with DoD policy.

DISA approved the STIG for Samsung Knox 2.x.

DISA Approved Product List

DISA has approved select Knox-enabled devices to the US DoD Approved Products List (APL).

***NOTE**: Select Samsung Knox-enabled devices and tablets are certified under the National Information Assurance Partnership (NIAP) Common Criteria (CC) Mobile Device Fundamental Protection Profile (MDFPP).*

Common Criteria Certification

The Common Criteria for Information Technology Security Evaluation, commonly referred to as Common Criteria, is an internationally-recognized standard for defining security objectives of information technology products and for evaluating vendor compliance with these objectives. A number of Governments use Common Criteria as the basis for their own certification schemes.

Select Galaxy devices with Knox embedded received Common Criteria (CC) certification. The current CC certification targets the new Mobile Device Fundamentals Protection Profile (MDFPP) of the National Information Assurance Partnership (NIAP), which addresses the security requirements of mobile devices for use in enterprise.

Samsung Knox is approved by the United States government as the first NIAP-validated consumer mobile devices to handle the full range of classified information.

Certifications

CSfC

Fifteen Samsung devices have been listed in the NSA/CSS's Commercial Solutions for Classified Program (CSfC) for approved security components.

ANSSI

Samsung Knox has obtained first-level security Certification Sécuritaire de Premier Niveau (CSPN) from the Agence nationale de la sécurité des systèmes d'information (ANSSI). The CSPN methodology and criteria is defined by ANSSI with evaluations run by ANSSI accredited testing labs.

ISCCC

Samsung Knox received the security solution certificate by the China Information Security Certification Center (ISCCC). Samsung worked closely with ISCCC to develop the certification process, including device requirements and security standards. By securing the critical ISCCC certification, Samsung has a stronger foothold to garner mobile device contracts with China's regulated industries, including government authorities, ministries, and finance.

CESG Approved

The Communications and Electronic Security Group (CESG) approved Knox-enabled Android devices for United Kingdom government use.

FICORA

Samsung devices with Knox fulfill national security requirements as defined by the Finnish National Security Auditing Criteria (KATAKRI II).

ASD

Australian Signals Directorate: ASD endorsing the Protection Profile for Mobile Device Fundamentals as well as recognizing evaluations against this Protection Profile.

Note: For the most recent updates to Samsung Knox certifications, see the following link:
https://www.samsungknox.com/en/security-certifications

About Samsung Electronics Co., Ltd.

Samsung Electronics Co., Ltd. is a global leader in technology, opening new possibilities for people everywhere. Through relentless innovation and discovery, we are transforming the worlds of televisions, smartphones, personal computers, printers, cameras, home appliances, LTE systems, medical devices, semiconductors and LED solutions. We employ 236,000 people across 79 countries with annual sales exceeding KRW 201 trillion. To discover more, please visit www.samsung.com

For more information about Samsung Knox, visit www.samsung.com/knox

Samsung Electronics Co., Ltd.
416, Maetan 3-dong, Yeongtong-gu
Suwon-si, Gyeonggi-do 443-772, Korea

Version	Date
An Overview of the Samsung Knox Platform V1.15	August 1, 2016
An Overview of the Samsung Knox Platform V1.14	Feb. 22, 2016
An Overview of the Samsung Knox Platform V1.13	Nov. 12, 2015

EXHIBIT 9

Chase and Samsung Pay®

Add your Chase card for a secure way to pay on-the-go.

Check out with Chase and Samsung Pay and enjoy the same protection and benefits you always get with your card.

Shop securely

When you use your Chase card with Samsung Pay, you can be confident knowing your purchases and account information are protected.

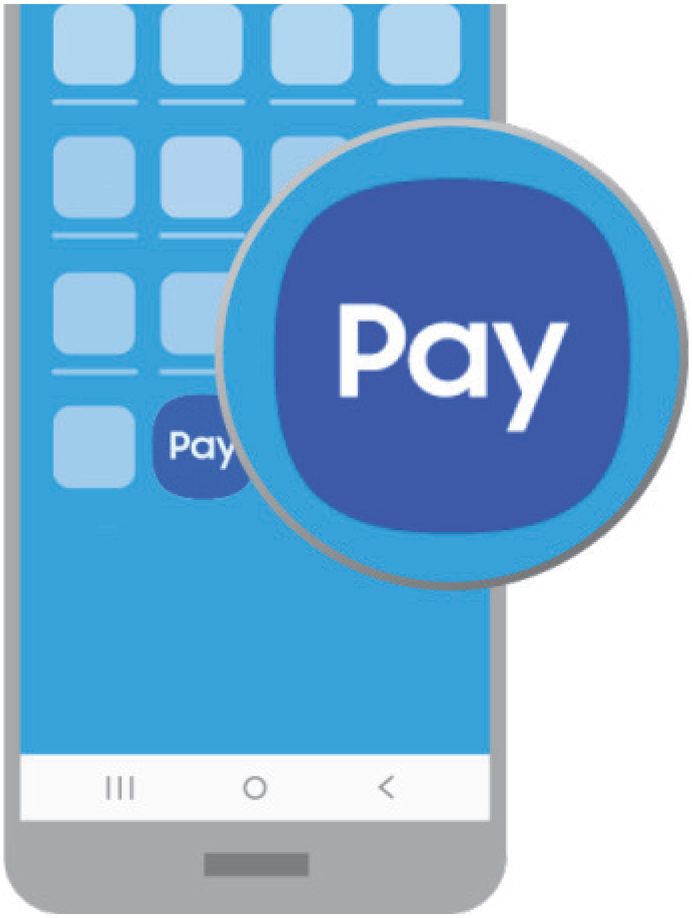
We help keep your account safe with **fraud monitoring technology** and Zero Liability.

Each transaction must be authenticated by your **fingerprint, PIN or iris scan**.

Your card number is never shared with the merchant or stored on your device.

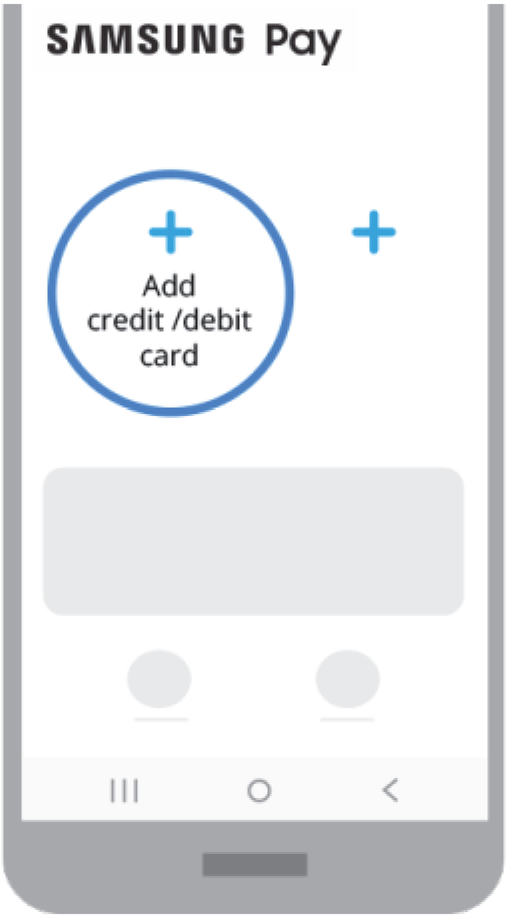
How to make your Chase card your default

Have your Chase credit or debit card handy — it only takes a few seconds.



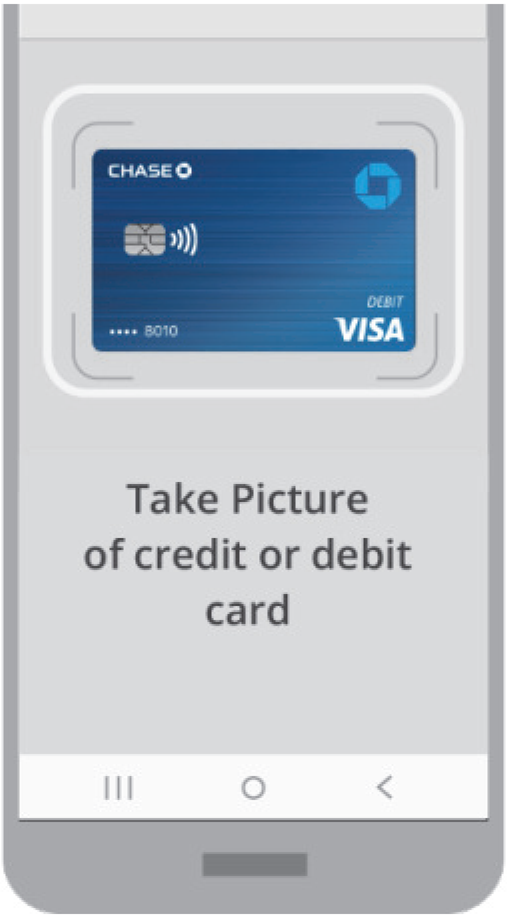
Open

Open Samsung Pay on your phone.



Choose

Choose "Add credit/debit card.”



Snap

Snap a pic of your card or enter the details manually.

Where and how to use Samsung Pay

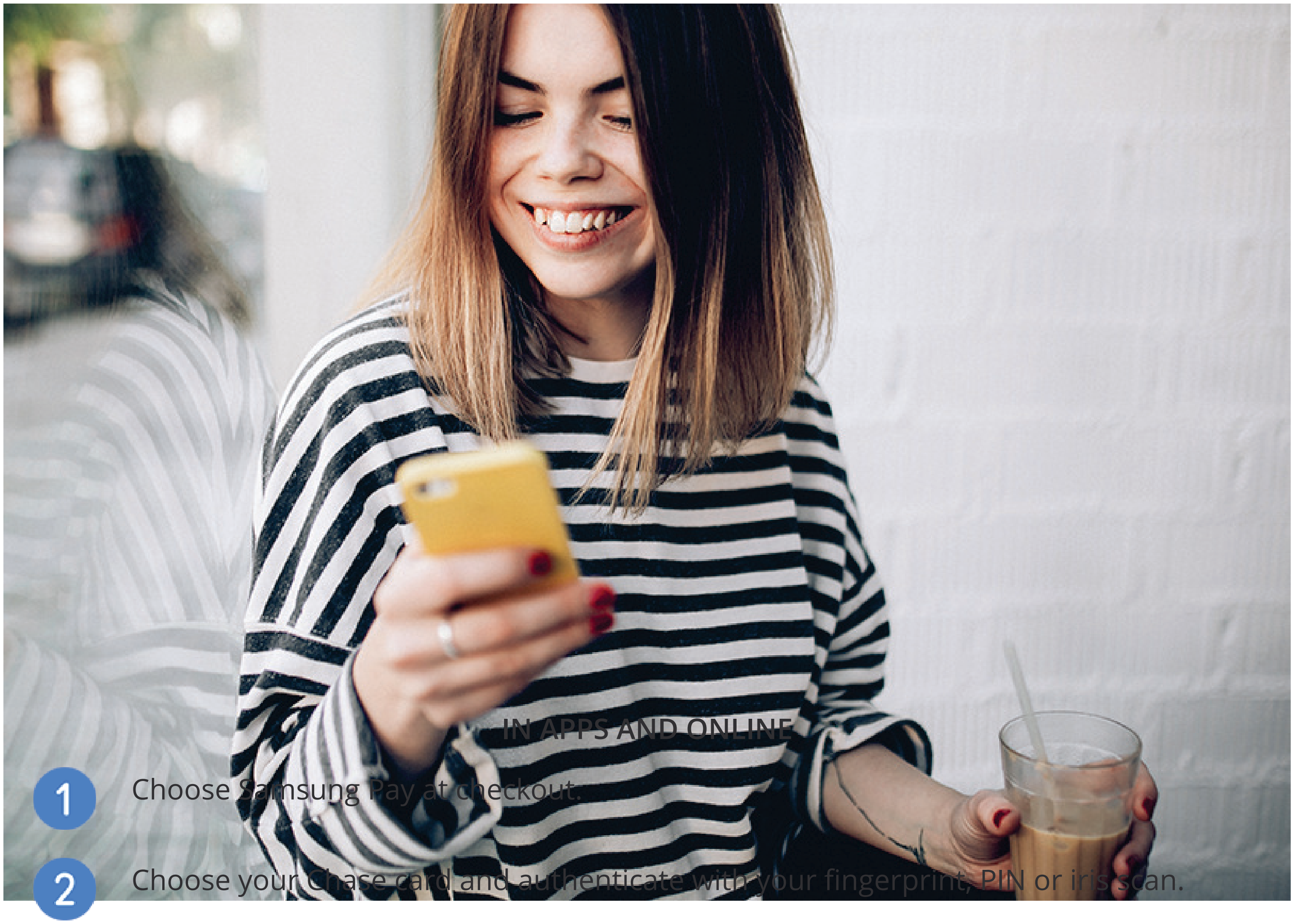
It's easy to use your Chase Visa® or Mastercard® with Samsung Pay in stores, in apps and online. Almost anywhere you can swipe or tap your card — in stores, in apps and online.

IN STORES

1 Open Samsung Pay and tap "Pay."

2 Choose your Chase card and authenticate with your fingerprint, PIN or iris scan.

3 Hold your phone near the card reader to complete the purchase.



IN APPS AND ONLINE

1. Choose Samsung Pay at checkout.

2. Choose your Chase card and authenticate with your fingerprint, PIN or iris scan.

CHASE Sign in

Look for the Cardless and Samsung Pay symbols at checkout

See stores and apps that accept Samsung Pay

Learn more about contactless payments

Accepted cards

Most Chase credit and debit cards are accepted in Samsung Pay.

See all accepted cards

General

What is Samsung Pay?

Samsung Pay provides owners of select Samsung Galaxy devices the ability to use their Galaxy phone to make payments almost anywhere you can swipe or tap your Chase credit, debit and/or Chase Liquid® cards.

How does Samsung Pay work?

Select Samsung Galaxy devices are equipped with Near Field communication technology (NFC) that transmits the tokenized version of the credit card / debit card payment credentials securely to a merchant terminal. In addition to NFC, Samsung Pay supports a new proprietary technology, Magnetic Secured Transmission (MST) which transmits the same secure tokenized credit card / debit card payment credentials. See below for more information about NFC and MST.

How do I get Samsung Pay on my Device?

Samsung Pay is available with the Galaxy Note8, S8, S7, S7 edge, S6, S6 active, S6 edge+, S6 edge, Note5, Gear S2 (with NFC only) and Gear S3 through an app update. The availability will vary by mobile network operator. The app will appear when the customers install the operating system update. Samsung Pay will be a preloaded app on upcoming Samsung Iconic devices. The app is not available for download on unsupported devices.

How do I add my Chase cards to Samsung Pay?

1. Choose the Samsung Pay icon, and log in using your Samsung account information.
2. If you do not already have a Samsung account, you can create one.
3. Once logged in you can, touch ADD CARD in Samsung Pay app to begin the process of adding your card.
4. Center the card in the on-screen window and Samsung Pay will read the card number. Or you can enter the card number manually.
5. Verify or enter the additional information such has Cardholder Name, Expiration Date and Security Code (CVV).
6. Accept the Terms and Conditions for adding a card to a digital wallet by touching AGREE TO ALL.
7. You may be prompted to select a delivery method for receiving a One Time Passcode, such as SMS or EMAIL or CALL IN.
8. The One Time Passcode will be delivered to the destination selected. Enter the code received and touch SUBMIT.

How do I pay using Samsung Pay?

1. Once your eligible Chase card is registered to Samsung Pay, you may access Favorite Cards by swiping up from the bottom of your screen.
2. The most recently added or used card is displayed. Swipe left or right to scroll through your registered payment cards.
3. When you have the desired card selected, place your finger on the **Home** button. To use your Samsung Pay **PIN**, touch **PIN**. Then, enter your four-digit **PIN**.

Samsung Pay will indicate that you are ready to make a payment.

For NFC Payments

1. Hold the phone above the NFC reader on the payment terminal
2. Align the camera with the NFC logo
3. The two devices should be almost touching

For MST Payments

1. Hold the phone closely to the terminal
2. Align the camera to face the magnetic stripe card reader
3. The two devices should be almost touching

Using Samsung Pay on the Gear S3

1. Simply press and hold the back key to launch Samsung Pay on your Gear S3
2. Rotate the bezel to select a card
3. Tap to pay at any NFC or MST terminal

If necessary, complete the transaction on the payment terminal. For example, if you are using a debit card, you are still required to enter your Personal Identification Number (PIN). Some merchants/terminals may prompt you to verify the total charges are correct, while others will require a signature.

Is there a fee associated with using a Chase card with Samsung Pay?

There is no cost to use Samsung Pay from Chase; however, an active data plan is required. Based on your wireless plan and mobile carrier's offering, additional message and data charges may apply. Your credit card terms and conditions, your debit card Deposit Account Agreement or your Chase Liquid agreement will apply for purchases. Chase does not charge any fees to add your credit, debit and/or Chase Liquid cards to Samsung Pay.

Can I use Samsung Pay outside of the United States?

charges and foreign transaction fees may apply.

Can a Samsung Pay token be used for recurring charges, subscriptions or bills (Netflix, Amazon Prime, etc)?

This service is not currently available.

What is Magnetic Secured Transmission (MST) Technology?

Near Field Communication (NFC) is a short-range wireless technology that allows two devices to exchange payment information quickly and conveniently at close proximity. Samsung Pay uses NFC technology or MST to transmit payment information from your phone to the contactless payment terminal.

Your Chase Card

Which Chase cards can I add to Samsung Pay?

Many Chase Visa® and Mastercard credit cards, debit and Chase Liquid cards work with Samsung Pay. Follow the instructions for adding your card to Samsung Pay (see Q&A: How do I add my Chase cards to Samsung Pay?) Call the number on the back of your card to see if it is eligible for Samsung Pay or check eligible Chase cards.

How does my Chase credit, debit or Chase Liquid card work with Samsung Pay?

When you add your Chase credit, debit or Chase Liquid card to Samsung Pay, the number on your card is replaced with a secure device account number (also called a token). This token is passed onto the merchant for payment instead of your actual card number.

Can I add my card to multiple Samsung Pay wallets?

You can add your card to multiple Samsung Pay wallets. However, for security purposes, your activation code will always be sent to the primary cardholder.

How do I remove my card from Samsung Pay?

When in the Samsung Pay app, you can touch your card to view the Card details, and also select the delete option.

Can I have more than one Chase card in Samsung Pay?

You can add up to 10 cards total to the Samsung Pay app. Any number of those could be eligible Chase cards.

Any eligible credit/debit card can be used with Samsung Pay, even chip/signature cards. Keep in mind that Samsung pay uses a digital version of your account, and is distinct to how you use your physical plastic cards, which remains unchanged.

Can I add the same Chase credit/debit cards that I have on my other mobile digital wallets to Samsung Pay?

You can add any eligible Chase credit, debit or Chase Liquid card into Samsung Pay, including cards that you may have in other digital wallets, such as Apple Pay® and Google Pay™.

Where can I use Samsung Pay?

Samsung Pay can be used with any merchant whose terminal can accept Near Field Communication (NFC), and because Samsung Pay also uses Magnetic Secure Transmission (MST), it can be used virtually anywhere you can swipe your card.

Security

Can I lock Samsung Pay?

Samsung Pay is locked when the device is locked. A PIN or Fingerprint is required per transaction in order to use Samsung Pay.

What is Hosted Cloud Emulation?

Host cloud emulation (HCE) is the software architecture that provides exact virtual representation of credit/debit cards using only software. On the other hand, other digital wallet services stores credit/debit card credentials on a secure element within the device.

What is a device account number (token)?

A device account number (token) is a substitute account number that replaces your card number in each transaction. This means that your actual information isn't shared when your shop and your details stay safe.

Am I liable for unauthorized charges made with Samsung Pay?

Purchases made with your Chase credit, debit and/or Chase Liquid cards have Chase Zero Liability Protection for any unauthorized card transactions. This means that you will not pay for any unauthorized transactions as long as they are reported promptly. Certain limitations apply. Please see your cardholder, deposit account and/or Chase Liquid card agreement for complete details.

Separate contact information exists for Credit Card and Personal Banking accounts.

- **Credit Card**: If your card has been lost, stolen or damaged, call us immediately at 1-800-432-3117. Go to chase.com/customerservice for call center hours.
- **Personal Banking**: If your card has been lost, stolen or damaged, call us immediately at 1-800-935-9935. Go to chase.com/customerservice for call center hours.

We accept operator relay calls. If you're deaf, hard of hearing, or have a speech disability, call 711 for assistance.

What if my device is lost or stolen?

If your phone is lost or stolen, you can suspend Samsung Pay by using the “Find My Mobile (FMM)” through Samsung or contacting Samsung directly. You can also call Chase, and we’ll assist you with suspending the cards in Samsung Pay, allowing you to continue to use your plastic cards.
If you find your phone, you can unsuspend Samsung Pay by following the prompts on your phone or through “FMM”.

What happens when I replace or update my device?

If you replace or update your device, deactivating your old device will delete the Digital Account Number (Token) associated with the old device. You'll then have to register your cards to your new device to use Samsung Pay, the same way you registered the first time.

What happens when my card is replaced?

Whether your card is replaced because it is lost or stolen, or if the card has expired, in most cases your new card will be automatically associated to the existing Digital Account Number (Token) in your device, and continue to be used for payments in Samsung Pay. If, for some reason, Chase could not re-add your new card to the existing Digital Account Number (Token), you'll see a notation in the Samsung Pay App that the card isn't active for use. You can then re-add your card, as you did initially to make it active again.

How do I make returns with purchases I’ve made using Samsung Pay?

You can make a return the same way you would with your plastic card today by contacting the merchant directly. We do recommend you bring the device that you used when returning items purchased with Samsung Pay.

CHASE

Sign in

Additional digital wallets

Google Pay

Apple Pay®

Other digital wallets

Esta página contiene información acerca del uso de tu tarjeta Chase Visa® y Mastercard en billeteras digitales.
Si tienes alguna pregunta, por favor, llama al número que aparece en el reverso de tu tarjeta.

©2020 Samsung Pay is a trademark of Samsung Electronics Co., Ltd. Use only in accordance with law.

Contact your bank or financial institution to verify that it is a Samsung Pay participant. Samsung Pay is available on select Samsung devices.

Only compatible with select cards and qualifying Samsung devices. Check with Chase to ensure that your card is compatible; and check the Samsung Pay Support page for additional compatibility information regarding devices, carriers and cards.

The listed merchant(s) are in no way affiliated with Chase, nor are the listed merchant(s) considered sponsors or co-sponsors of this program. All trademarks are the property of their respective owner(s).

Zero Liability Protection does not apply to use of an account by an authorized user without the approval of the primary cardmember. If you think someone used your account without permission, tell us immediately by calling the Cardmember Services number on your card or billing statement.

Chase, Chase Pay, and the Chase Octagon are trademarks of JPMorgan Chase Bank, N.A.

Amazon, the Amazon.com logo, the smile logo, and all related logos are trademarks of Amazon.com, Inc. or its affiliates.

Deposit and credit card products provided by JPMorgan Chase Bank, N.A. Member FDIC.

The Contactless Symbol and Contactless Indicator are trademarks owned by and used with the permission of EMVCo, LLC.

Follow us:

Use your mobile device or laptop to pay quickly and securely with your Chase Visa® and Mastercard.

Chase Digital Payments

Use your Chase Visa® and Mastercard® across a variety of devices to quickly and securely make purchases – all while still receiving the same great services and benefits Chase provides.

PayPal®

Chase with PayPal is an easy way to shop or make online payments with your Chase card. You can also use your Ultimate Rewards® to pay at checkout.

Apple Pay®

Chase with Apple Pay offers a simple, secure and private way to make purchases in stores, within apps and online.

Google Pay™

Chase with Google Pay enables you to pay your way. Wherever you are — at a store, in-app or online — you're ready to use Google Pay

Samsung Pay

Chase with Samsung Pay provides a safe and simple way to pay for purchases almost anywhere you can swipe or tap your card.

Other digital wallets

“Chase,” “JPMorgan,” “JPMorgan Chase,” the JPMorgan Chase logo and the Octagon Symbol are trademarks of JPMorgan Chase Bank, N.A. JPMorgan Chase Bank, N.A. is a wholly-owned subsidiary of JPMorgan Chase & Co.

About Chase J.P. Morgan JPMorgan Chase & Co. Privacy Security Terms of use

Our commitment to accessibility Site map AdChoices Member FDIC Equal Housing Lender

EXHIBIT 10

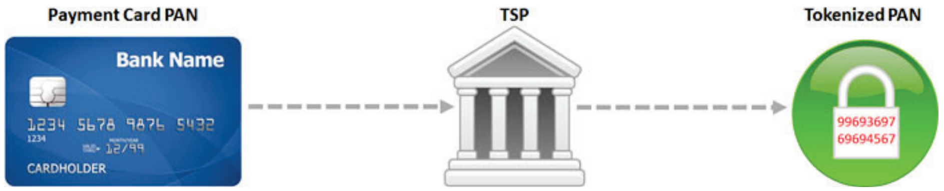
Code Lab
Mobile Tech Insights
Samsung Pass
S-Pen
Vulkan
Health
KNOX
eSIM
5G
Gear VR
Samsung Pay
Samsung Pay Security
Tokenization
Token Handling by Samsung Pay
Device-side Security: Samsung Pay, TrustZone, and the TEE
User Identity Setup and Credential Verification
Secured Communication with the Payment Networks
Remote Management
Conclusion
Early Access
Developer Events
Blog

Program › Mobile Tech Insights › Samsung Pay › Tokenization

Tokenization

What it is and why it’s important

Tokenization is the process of replacing essential credit card credentials — the 16-digit primary account number (PAN), for instance — with a substitute value. Called a token PAN or digitized PAN (DPAN), the token protects the real card number from theft and misuse. Payment tokenization adds a cryptogram to the mix. The cryptogram contains unique authentication data generated by the smartphone device. It demonstrates to the card network that the device and the card being used are genuine and not a vehicle for intercepted or cloned credentials. Moreover, the tokens transmitted from the point of sale (POS) can only be tied back to sensitive information kept on highly secure servers maintained by the token service provider (TSP). Samsung Pay currently utilizes the tokenization services offered by the global payment networks, which are available to all respective card association members, although third-party TSP integration is also supported, as well as TSPs independently owned/operated by card issuers themselves.



The card issuer sets the rules and parameters of the token service, conducts account verification and cardholder authorization during the token request stage, and authorizes transactions. In the event of a breach, tokens are of no use, and the payment data is kept secure from hackers. Tokens do not replace EMV (Europay, MasterCard, and Visa — the three companies that originally created the chip-and-PIN standard), but are a chip-and-PIN alternative for mobile payments. For more on EMVCo and the specifications for contact, contactless, mobile, and payment tokenization, visit https://www.emvco.com .

Token request and issuance

A TSP offers token requester (TR) registration, token lifecycle management, security and control, and processing management. Before a token is issued, the mobile wallet provider — in this case, Samsung, as the publisher of Samsung Pay — must first register with the card issuer’s designated TSP.

Once registered with the TSP, the TR can legitimately request a token on behalf of a cardholder and a specific device. This happens when the cardholder enrolls a card in Samsung Pay, at which point the request is sent by the TR to the TSP. Upon receipt of the request, the TSP performs a series of security controls and identification and verification (ID&V) processes with the card issuer via the payment network before issuing the token.

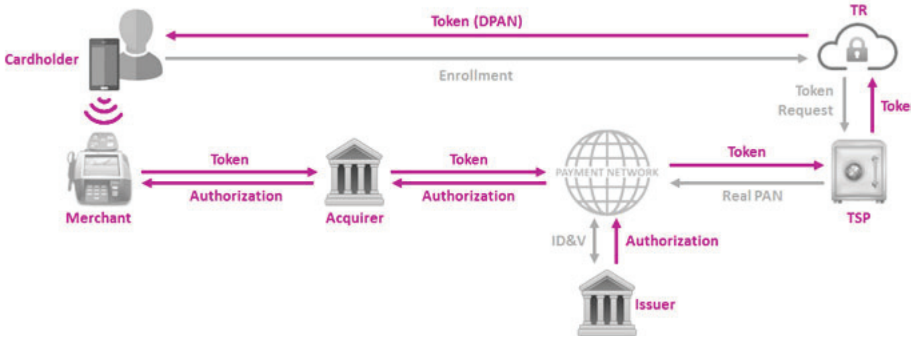


Figure 1. Tokenization within a mobile payments infrastructure

The tokenized DPAN is tied to the user's funding primary account number (FPAN) with the issuer, rather than to the physical PAN on the plastic card. Tokens remain active for as long as the account is active or is canceled, even if the plastic card expires or is canceled.

The user does not have to re-enroll the card when they receive a replacement card because the FPAN remains the same. In the event the card is lost, an alternate FPAN, tied to the existing DPAN, is issued. Meanwhile, the DPAN has its own expiration date. When the expiry date is reached, a new set of keys are replenished. The DPAN is not reprovisioned, nor does it require re-enrollment of the physical card.

Cryptographic key generation

The cryptogram is another important element ensuring a transaction's integrity. It contains encrypted data derived from the token (DPAN), timestamp, and Application Transaction Counter (ATC), which are used to prevent a "replay" event — repeating a transaction using the same authorization code.

When making a purchase, the user's device sends the payment token, along with a cryptogram, to the merchant POS, which relays them to the payment network for approval by the issuer. The cryptogram is generated using a cryptographic key based on the algorithm furnished by the card network. See Token handling by Samsung Pay for additional details.

This key, stored in the device's trusted execution environment (TEE), called TrustZone, is **static** or **dynamic**, depending on the card brand/network. Static cryptographic keys are used over a relatively long period of time and in multiple key exchanges, whereas a dynamic key is generated for each exchange.

Multiple dynamic keys, often called limited use keys (LUK), are provisioned at card enrollment. The number of keys provisioned is regulated by the payment network. Each time Samsung Pay is used to make a credit card transaction, one key is consumed. Keys are replenished based on card network replenishment logic residing in both the software development kit (SDK) of the card network and the card network backend.

The card networks hold the master key to their card product and use it to generate a unique derived key (UDK) for each cardholder, which remains unchanged for the lifetime of the card. Once the cryptogram—generated using the static/dynamic keys in the device—is verified by the TSP on behalf of the issuer, transaction processing continues. If the cryptograms don't match, the transaction is flagged.

Rules governing tokens and keys

Issuers are responsible for providing the card keys to the TSP, which sets the rules in accordance with the respective payment network's specifications governing the number of transactions, value, channel, and valid timeframe within which the key can be used. The specific details of key management are governed by the payment networks and are different for each. Typically, the function of key management is undertaken by the TSP or a third party.

In the case of payment networks mandating dynamic keys after provisioning, the user's device must obtain the keys ahead of time. Invisible to the user, multiple keys are downloaded to the device during a process called "replenishment." Once keys are used or expire, the device must be online (connected to the Internet via Wi-Fi or wireless carrier) to obtain new keys for future transactions.

As discussed in TEE-based token management, hardware-based secure storage on Samsung devices makes static keys the preferred key type. Regardless of key type —static or dynamic — both the correct token and a correct key must be present to successfully process a payment.

Domain restrictions ensure the token is used for the intended cases for which it is approved. These restrictions specify the token's use case, channel, and merchant for each TR. A token is only valid when it originates from the device, channel, merchant, TR, and use case for which it is intended. Token domain restrictions are defined by the TSP during the TR registration process.

Identity and Verification (ID&V) and token assurance

In addition to supporting the authorization decision by the card issuer, the token assurance level indicates to merchants, acquirers, and processors the present degree of confidence in the payment token to PAN-cardholderdevice binding. It is determined by the outcome of the ID&V steps conducted at the time the token is issued. An issuer's chosen ID&V methods can range from no ID&V to a combination of user-supplied data, which can include billing address, device ID and location, and various communication channels—3D secure login, mobile banking, federated login, and one-time password (OTP). OTP is the most common ID&V channel. Samsung Pay currently supports OTP via SMS and email, call center, and app-to-app channels.

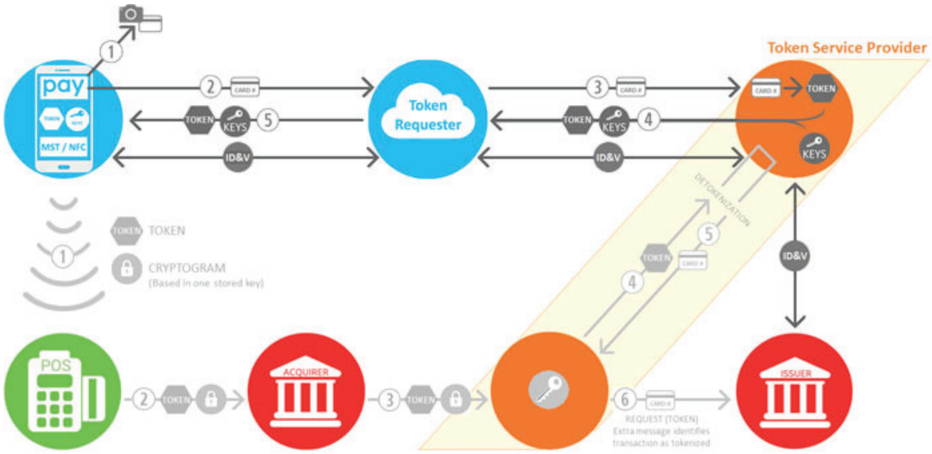


Figure 2. Card Enrollment and Payment Workflows

Privacy Policy Privacy Policy (EU) Terms and Conditions End User License Agreement

Bixby

Events

Blog

Blog Articles
Blog Insights

SmartThings

Services & APIs

CAPH 3.0 for Smart TV
Galaxy Add-on SDKs
Game
Gear 360 SDK
Galaxy Watch Designer
In-App Purchase SDK
NaCl for Smart TV
PEN.UP
Samsung DeX
Samsung Health
Samsung Internet
Smart TV Product API
Samsung Stickers
Samsung Themes
Smart View SDK
TEEGRIS SDK
Tizen Extension APIs for Galaxy Watch
TOAST

Devices

ARTIK IoT Modules
Family Hub
Galaxy
Galaxy Watch
Smart TV
Z (Tizen Mobile)

Program

Developer Program
Made for Samsung
Galaxy Apps
Development Guides
Remote Test Lab
Code Lab

Community

Forum
- Family Hub
- GALAXY
- Smart TV
- Z (Tizen Mobile)
- Services
- Open Talk
- Galaxy Watch

Support Dashboard
SEED Program

SAMSUNG

EXHIBIT 11

Code Lab
Mobile Tech Insights
Samsung Pass
S-Pen
Vulkan
Health
KNOX
eSIM
5G
Gear VR
Samsung Pay
Samsung Pay Security
Tokenization
Token Handling by Samsung Pay
Device-side Security: Samsung Pay, TrustZone, and the TEE
User Identity Setup and Credential Verification
Secured Communication with the Payment Networks
Remote Management
Conclusion
Early Access
Developer Events
Blog

Program › Mobile Tech Insights › Samsung Pay › Token Handling by Samsung Pay

Token Handling by Samsung Pay

Leveraging the Samsung KNOX security framework [1]

Samsung Pay's deep integration with the Samsung KNOX platform for device-side security is the architectural attribute that immediately differentiates it from every other mobile wallet app. Pivotal to Samsung KNOX security is its TEE, ensuring that the user's personal identity is safe while reliably providing issuers with the information necessary to make accurate risk assessments.

All Samsung Pay tokens and keys are stored within the TEE in encrypted form using a hardware-based device key that is unique to each device. Device tampering or the introduction of malware invoke preventive safeguards that include disabling the app and even shutting down the device.

> Note All cryptographic methods and handling performed by Samsung Pay are implemented in accordance with the specifications provided by the respective card network and the designated TSP.

During card provisioning, the TSP replaces the 16-digit PAN on the credit or debit card with a 16-digit substitute (DPAN) and relays it through the TR to the user's device. Then, whenever the device is used to make a purchase, Samsung Pay generates a cryptogram from the applicable card network algorithm and transmits it to the merchant's POS upon cardholder authentication, either by fingerprint scan or by entering the correct Samsung account PIN.

Remember, tokens can only be tied back to sensitive cardholder information kept in the "vault" maintained by the TSP. In the event of a breach — because the true PAN is not stored on the device and therefore is never passed to the POS — tokens are of no use to thieves and payment data is kept secure from hackers and other threats.

When Samsung Pay transactions are processed through a payment network, the TSP is called to convert tokens back into a PAN to allow the issuer to process the transaction in the normal way.

Four primary processes involve Samsung Pay: card enrollment/token provisioning, transaction processing, token replacement/replenishment, and token suspension/resumption/deletion. The participating parties are cardholder, merchant, acquirer, payment network, TSP, and issuer. Samsung Pay's token request (TR) service has an active role in the token provision/replace/replenish and suspend/resume/delete workflows.

Samsung Pay currently supports two types of token provisioning/replenishment models or methods: cloud-based and TEE-based.

Cloud-based key management

As discussed previously, cloud-based token management employs dynamic keys called limited use keys (LUK) or limited use payment credentials (LUPC), of which a fixed number are initially generated at provisioning (for example, 5 or 20). For card brands employing a cloud-based mobile payment (CBMP) system, a provisioned key is consumed each time the corresponding card enrolled in Samsung Pay is used to make a purchase, or a provisioned key is used for a limited number of transactions. Keys are replenished by the TSP based on the number of keys already used, the number remaining, and time to live, but this can only occur when the device is online — connected to the Internet via Wi-Fi or wireless carrier. If all available use keys on the device are consumed while the device is offline —and consequently unable to communicate with the TR server and, by extension, the TSP — subsequent transactions are not recognized. However, once the device comes back online and connectivity is restored, keys are replenished and normal transaction processing resumes.

TEE-based key management

By contrast, the TEE-based key management model employs a static key and is used for multiple transactions, as many as the issuer authorizes based on the account's available credit/funds until suspended or deleted, either by the cardholder or the issuer. Like dynamic keys, this static type of key is securely stored in the TEE of Samsung devices supporting Samsung Pay. During transaction processing, it is used to generate the cryptogram containing the token on file with the TSP.

And, because this type of key is stored in the TEE, cryptograms can be generated on demand so users can make purchases whether the device is online or offline. There is no need for key replenishment. As necessary, TEE-based keys are replaced (rather than replenished), and they can be suspended and/or deleted just like their cloud-based counterparts.

As a value proposition, this translates to lower cost and greater end-to-end flexibility for integrating a wider variety of use cases—loyalty cards, coupons, gift cards, and the like—in addition to credit and debit cards because specific transactions can be mapped to the cardholder.

1. See White Paper: Samsung KNOX Security Solution for a look at the full array of Samsung KNOX security features.

Privacy Policy Privacy Policy (EU) Terms and Conditions End User License Agreement

Bixby

Events

Blog

Blog Articles
Blog Insights

SmartThings

Services & APIs

CAPH 3.0 for Smart TV
Galaxy Add-on SDKs
Game
Gear 360 SDK
Galaxy Watch Designer
In-App Purchase SDK
NaCl for Smart TV
PEN.UP
Samsung DeX
Samsung Health
Samsung Internet
Smart TV Product API
Samsung Stickers
Samsung Themes
Smart View SDK
TEEGRIS SDK
Tizen Extension APIs for Galaxy Watch
TOAST

Devices

ARTIK IoT Modules
Family Hub
Galaxy
Galaxy Watch
Smart TV
Z (Tizen Mobile)

Program

Developer Program
Made for Samsung
Galaxy Apps
Development Guides
Remote Test Lab
Code Lab

Community

Forum
- Family Hub
- GALAXY
- Smart TV
- Z (Tizen Mobile)
- Services
- Open Talk
- Galaxy Watch

Support Dashboard
SEED Program

SAMSUNG

EXHIBIT 12

Code Lab

Mobile Tech Insights

Samsung Pass

S-Pen

Vulkan

Health

KNOX

eSIM

5G

Gear VR

Samsung Pay

Samsung Pay Security

Tokenization

Token Handling by Samsung Pay

Device-side Security: Samsung Pay, TrustZone, and the TEE

User Identity Setup and Credential Verification

Secured Communication with the Payment Networks

Remote Management

Conclusion

Early Access

Developer Events

Blog

Program › Mobile Tech Insights › Samsung Pay › Device-side Security: Samsung Pay, TrustZone, and the TEE

Device-side Security: Samsung Pay, TrustZone, and the TEE

Worlds apart from other wallet apps

Samsung's Galaxy-class devices supporting KNOX and Samsung Pay employ ARM® TrustZone® technology , a system-on-chip (SoC) security architecture that establishes two hardware-based "worlds" — a **Normal** World and a **Secure** World. The Normal World is where non-secure software and data processing takes place. The Secure World is reserved for storage and computing of sensitive (encrypted) data and the associated cryptographic keys.

By erecting a strong security perimeter between the two worlds, hardware logic present in the TrustZone bus fabric prevents Normal World components from accessing Secure World resources. Pictured in Figure 3, the TrustZone monitor controls switching between worlds. Applications that run in the Secure World are called Trusted Apps(TAs).

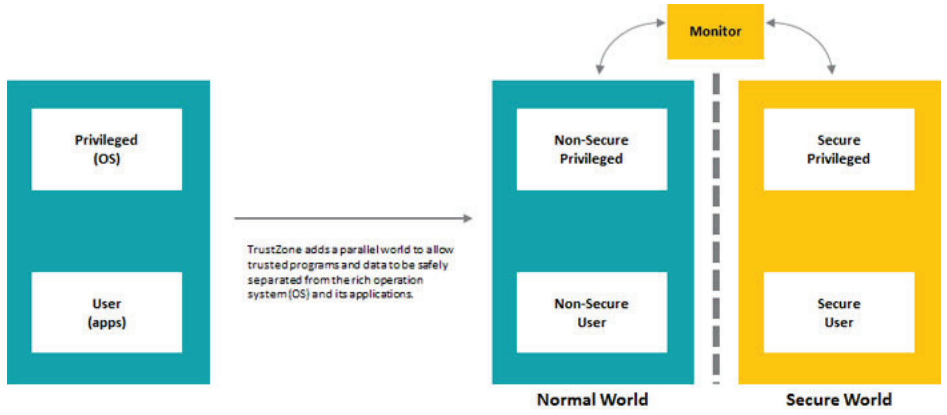

Monitor
Privileged (OS)
User (apps)
TrustZone adds a parallel world to allow trusted programs and data to be safely separated from the rich operation system (OS) and its applications.
Non-Secure Privileged
Non-Secure User
Normal World
Secure Privileged
Secure User
Secure World

Figure 3. TrustZone creates two parallel execution worlds

The combination of TrustZone-based hardware isolation, Trusted Boot and a trusted OS make up the TEE on Samsung devices running Samsung Pay (Figure 4).

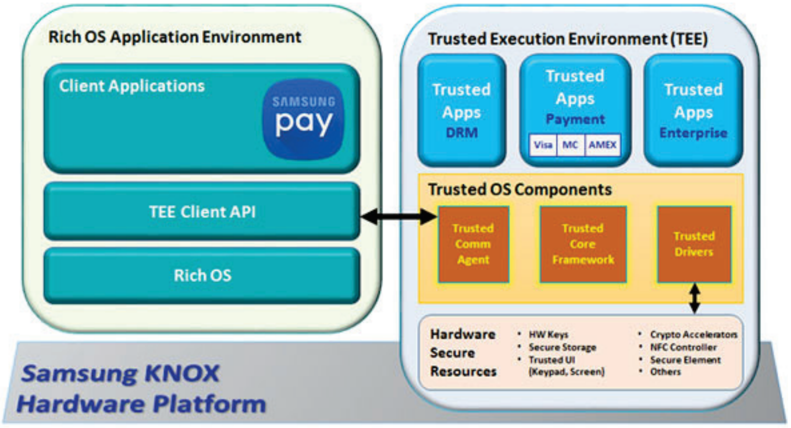


Figure 4. Trusted Execution Environment (high-level view)

Shown in Figure 4, multiple TAs comprising the Samsung Pay architecture, such as those responsible for communications with the payment networks, run inside the TEE. There are others as well, including the trusted apps that handle user authentication and those responsible for managing data encryption and authentication keys for the Payment Framework. For user authentication, trusted drivers operating in the TEE control access to the fingerprint sensor and the touch sensor for the Trusted PIN Pad. These drivers only allow authentication information to be passed directly to the respective payment network trusted app (for Visa, MasterCard, American Express, et al) inside the TEE.

Included with its cryptographically signed certificate, each TA is given a lifetime unique identifier. Communication restrictions between trusted drivers and trusted apps are based on a whitelist of these identifiers. The whitelists, managed by the trusted drivers themselves, cannot be modified.

In addition to the NFC controller, an MST antenna enables transmission between Samsung Pay and mag stripe POS card readers. Like the fingerprint scanner and the touch sensor, use of the MST antenna is guarded by a trusted driver, which restricts access to the TAs for payment networks only. Moreover, only the card track data authorized by the corresponding payment network is passed to the merchant POS by the respective trusted app.

A number of other defense-in-depth measures come with the KNOX framework to ensure comprehensive application, OS, and device integrity, including:

TrustZone-based Integrity Measurement Architecture (TIMA)

TIMA is a unique feature on Samsung mobile devices. As previously discussed, TrustZone hardware effectively partitions memory and CPU resources into a “secure” and a “non-secure” world. TIMA, running in the Secure World, uses the TrustZone hardware to continuously monitor the integrity of the Linux kernel. Along with Secure Boot and Security Enhancements for Android (SE for Android), TIMA forms the first line of defense against malicious attacks on the kernel and core bootstrap processes. If kernel or boot loader integrity violations are detected, TIMA takes a policy-driven action in response, one of which is to disable the kernel and restart the device to a known good state, thereby safeguarding all TIMA-dependent features within the TEE, including Samsung Pay and the Samsung KNOX Workspace, from device-level attacks.

Secure Boot, Trusted Boot, and remote attestation

During the device boot process, each of the bootloaders, the TEE, and the hardened Android kernel are verified through code signing. Most importantly, only the Samsung-approved TEE hosting the security-critical payment data and operations of Samsung Pay can be loaded to the devices. This safeguard is called **Secure Boot**.

In addition to Secure Boot, Samsung devices employ **Trusted Boot** to measure and record the cryptographic fingerprints of the bootloaders, the TEE, and the Android kernel. Then, during the provisioning of payment credentials, the Samsung Pay server remotely verifies the integrity of these key pieces of system software — particularly the TEE — using remote attestation. If any one of these elements has been modified, payments credentials are not provisioned to the device.

Trusted Apps verification

Whenever a trusted app is loaded into memory, the TEE performs cryptographic verification of the binary — the app’s executable program — to further ensure that only authentic Samsung Pay TAs are executed and allowed to access payment credentials. This check is performed in addition to the initial verification performed when the Samsung Pay app is first installed on the device.

Mandatory Access Control (MAC)

By employing SE for Android, Samsung Pay enforces MAC to ensure that only the authentic Samsung Pay app is allowed to execute Samsung Pay-specific functionalities, thereby restricting access to trusted apps only.

SE for Android stops mobile apps from granting themselves extra privileges, prevents apps from sharing too much data, and prevents the bypass of security features.

User authentication

In Samsung Pay, user authentication is handled by either the fingerprint scanner or the trusted PIN pad (TPP), both of which reside in TrustZone. Only when authentication is successful is the result (the verdict — affirmative user authentication) securely transferred to the respective payment network TA, which then talks to the tokenized NFC or MST trusted app interface to execute the payment. All requests for authentication and responses specifying the verdict are encrypted with keys known only to the intended TA recipients, all of which happens securely within the TEE.

Authentication verdicts are immediately cleared after transmittal to prevent any single user authentication from being used to attempt multiple payments.

Privacy Policy Privacy Policy (EU) Terms and Conditions End User License Agreement

Bixby

Events

Blog
Blog Articles
Blog Insights

SmartThings

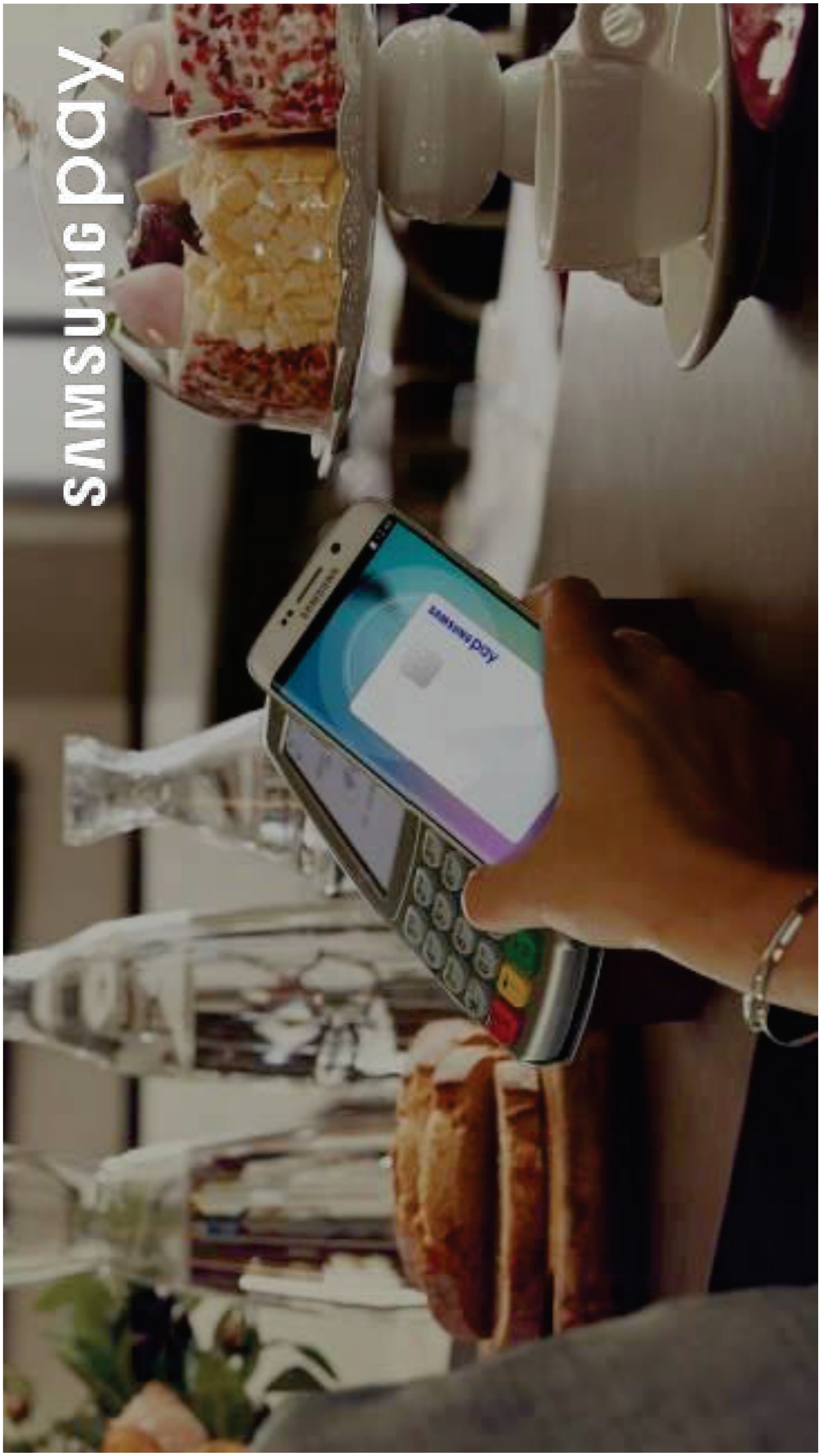
Services & APIs
CAPH 3.0 for Smart TV
Galaxy Add-on SDKs
Game
Gear 360 SDK
Galaxy Watch Designer
In-App Purchase SDK
NaCl for Smart TV
PEN.UP
Samsung DeX
Samsung Health
Samsung Internet
Smart TV Product API
Samsung Stickers
Samsung Themes
Smart View SDK
TEEGRIS SDK
Tizen Extension APIs for Galaxy Watch
TOAST

Devices
ARTIK IoT Modules
Family Hub
Galaxy
Galaxy Watch
Smart TV
Z (Tizen Mobile)

Program
Developer Program
Made for Samsung
Galaxy Apps
Development Guides
Remote Test Lab
Code Lab

Community
Forum
Family Hub
GALAXY
Smart TV
Z (Tizen Mobile)
Services
Open Talk
Galaxy Watch
Support Dashboard
SEED Program

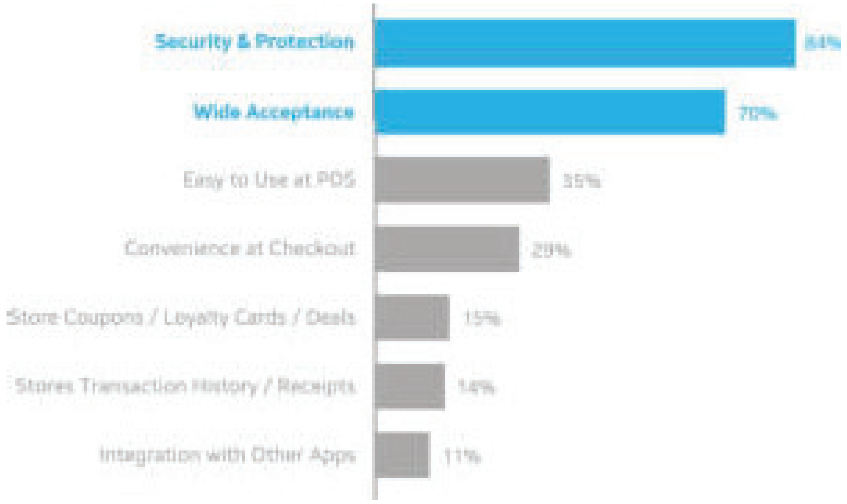
EXHIBIT 13


SAMSUNG pay

SAMSUNG pay

M-Payment Scenario

Most important factors to the end-user in a mobile payment app

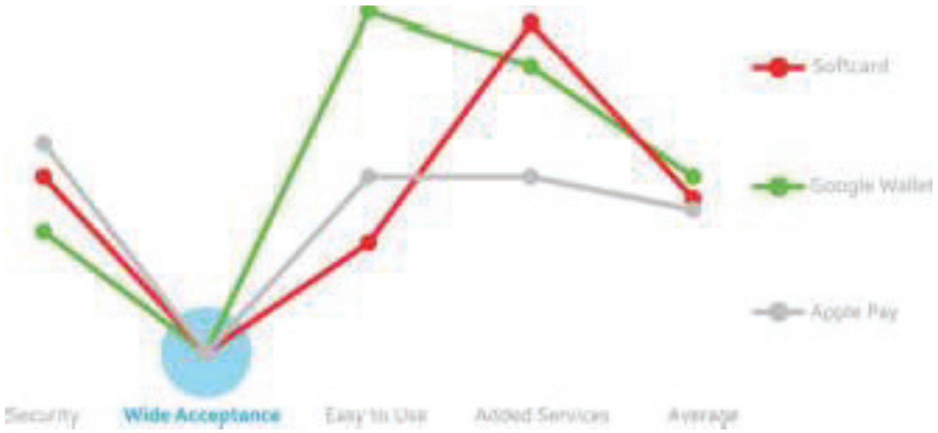


* The data above is based on market research conducted by Samsung.

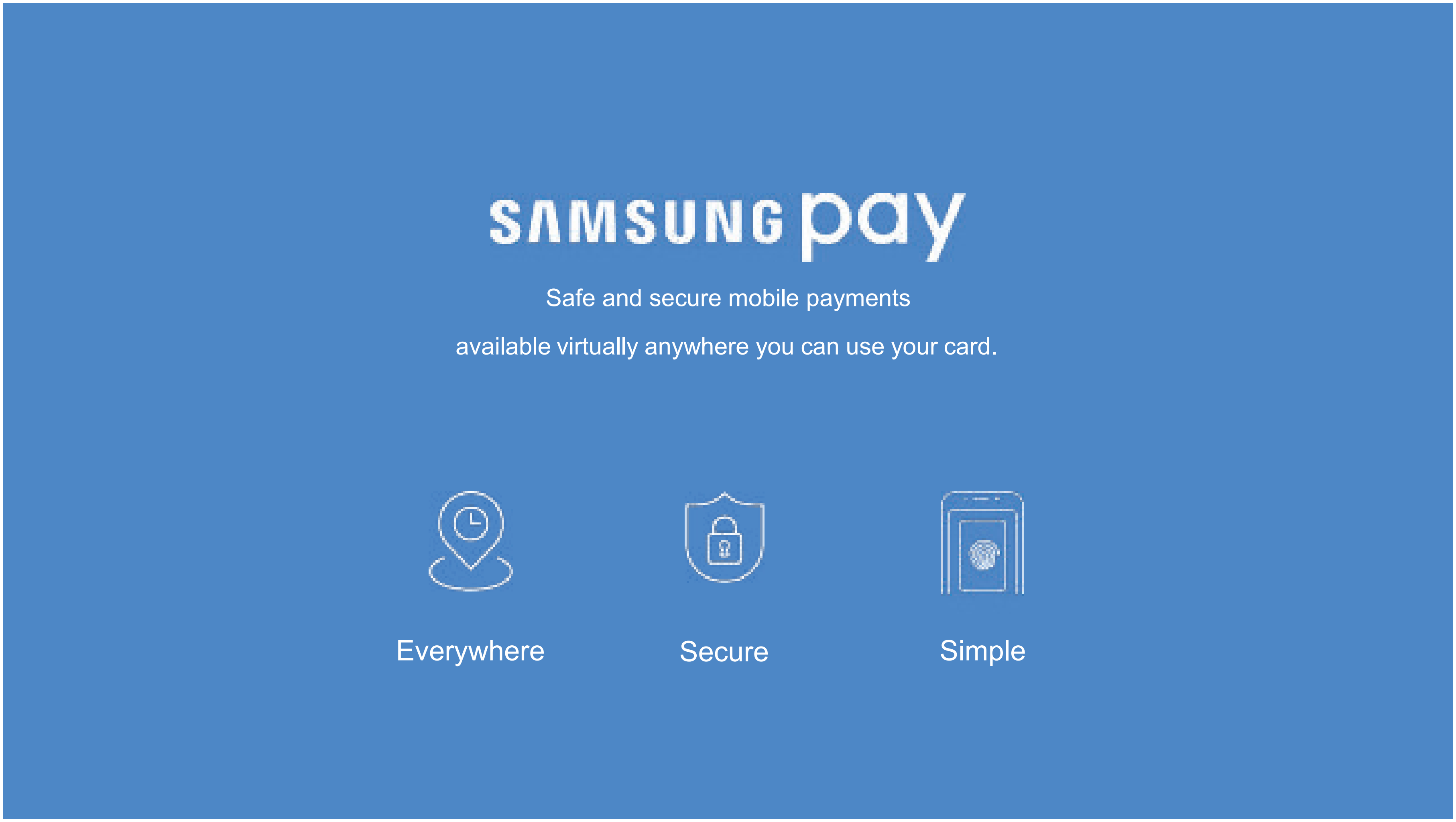
SAMSUNG pay

M-Payment Scenario

Wide acceptance is the biggest barrier of adopting m-payments across solutions



* The data above is based on market research conducted by Samsung



SAMSUNG pay

Safe and secure mobile payments

available virtually anywhere you can use your card.

SAMSUNG pay

Everywhere – Widely Accepted

Works almost everywhere cards are accepted

SAMSUNG pay

Secure - Device

Designed with our highest level of security available.

User Authentication

Samsung KNOX

TrustZone

SAMSUNG pay

Secure - Data

Designed with our highest level of security available.

Tokenization

Personal Data Protection

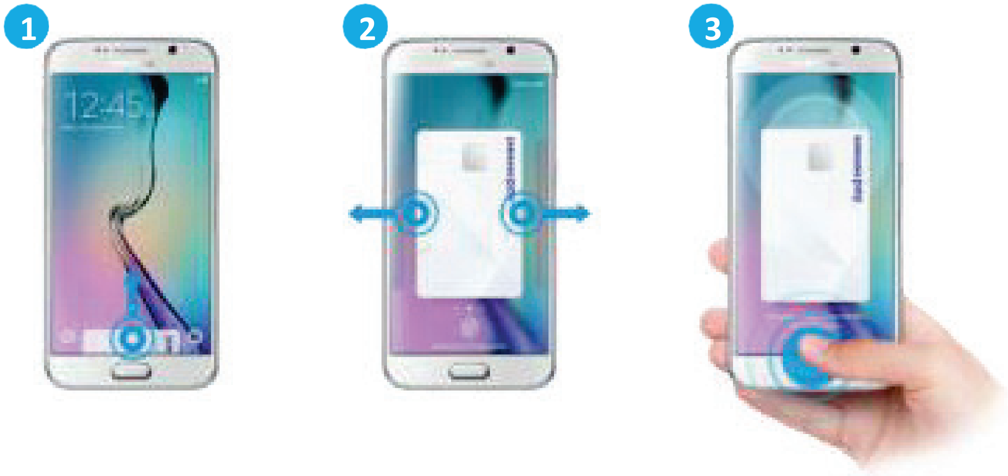
Remote Wipe

PII = Personal Identifiable Information

SAMSUNG pay

Simple

Designed to make paying with your phone fast, easy, and convenient.



SAMSUNG pay

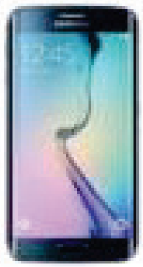
Eligible Devices

A510 & A520

A710 & A720

A910

G920* & G925*



G928

G930 & G935

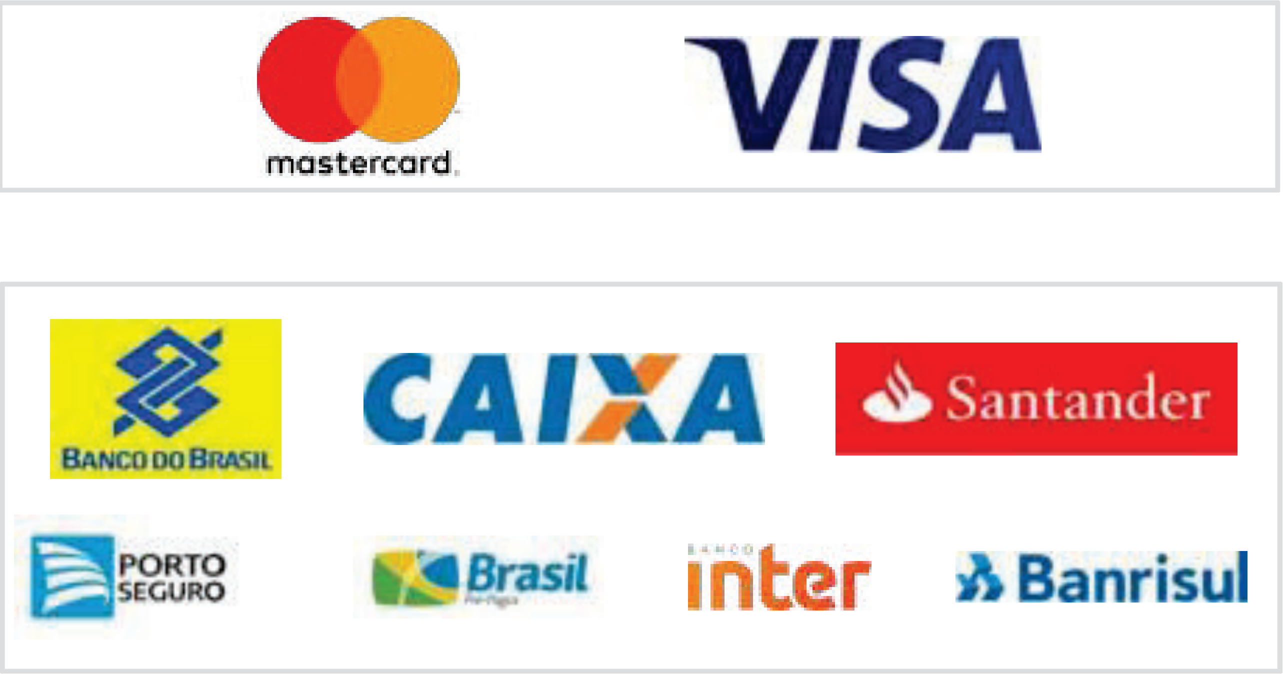
N920

G950 & G955

* NFC only

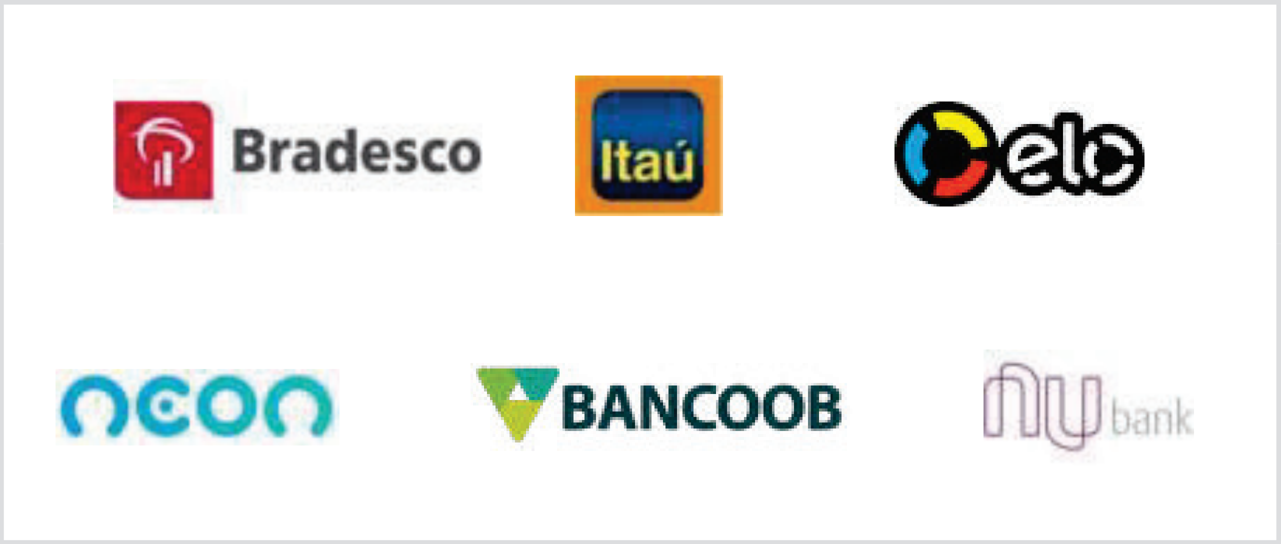
SAMSUNG pay

Partners - Launched



SAMSUNG pay

Partners - Coming Soon



SAMSUNG pay

Market comparison

Software provider	Monetization	Safety	Offline (in-store) Range	Online (in-app/ e-commerce) Range	Consumer base Potential	Expansion Potential
SAMSUNG Pay	● No ISSUER transaction fee	● TEE; Attestation; Finger Print; Token; Knox	● NFC; MST	●	◔ → ● Premium → Premium + Mid Q2'17	●
Pay	◔ ISSUER Transaction fee	● TEE; Finger Print; Token	◑ NFC	●	◔ Premium	◑
pay	● No ISSUER transaction fee	◑ HCE; Token	◑ NFC	●	◑ Android Users	●
PayPal pagseguro	◔ Transaction Fees	◔	○ None	●	◕ All	◔

Value Added Services

Issuer Apps Integration

Allows user to open issuer's bank application from Samsung Pay and also app-to-app ID&V

Membership and PLCC

Allows users to include personal (insurance, health care, etc), membership, PLCC and co-branded *cards*

Digital Promotions

Partner's promotion, push-notifications and geofence campaigns

Smartwatch

Payments thought Samsung Gear smartwatch

Transit

Include transportation cards to be used on subways, trains and buses

Thank you!

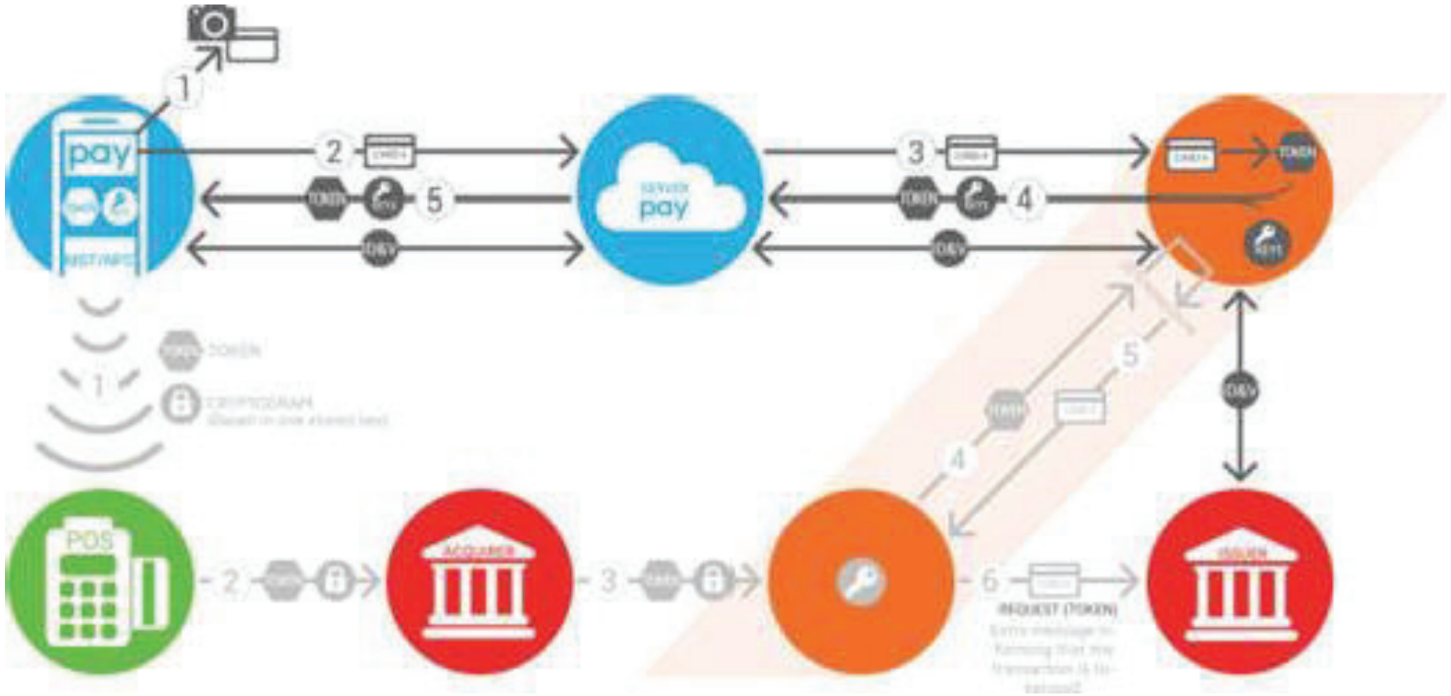
Obrigado

감사합니다

Appendix

SAMSUNG pay

Process Overview



CONFIDENTIAL MATERIAL

System Overview

Card Provisioning Mechanics

Sign-up

- User signs up for Samsung Pay using their Samsung Account.
- Card information is immediately encrypted and securely sent to the appropriate credit card network.

Tokenization

- Upon determining card validity, account info and device integrity, Network & Issuer send a **Token** to device.
- Token is stored in Trusted Execution Environment on the device, leveraging Samsung KNOX's Architecture.
- The Token (known as the Digital Card Number) is used in place of an actual credit card number.

Policy

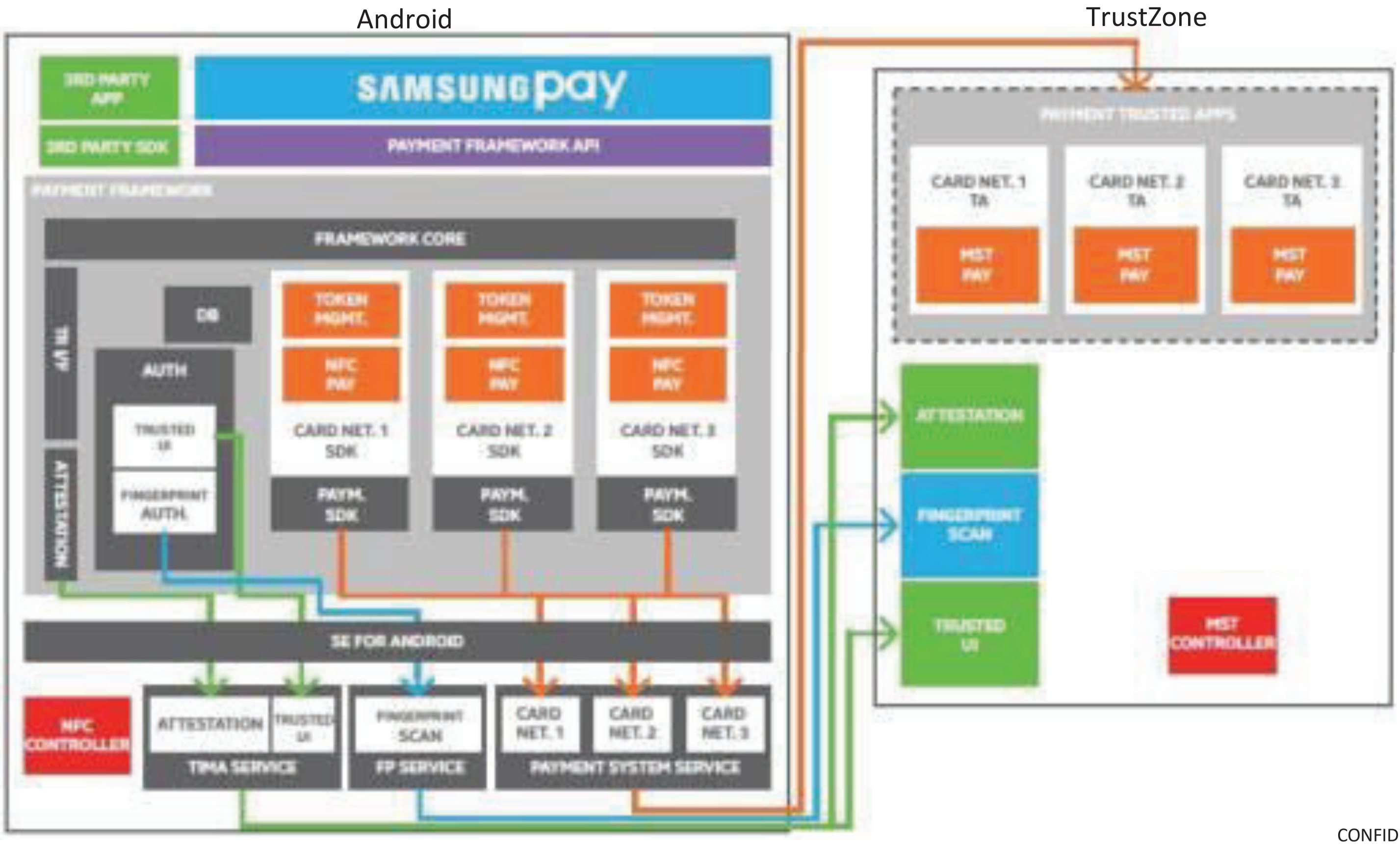
- No Credit Card Information is stored on our devices or servers.

Payment Mechanics

Security Methodology

- At time of transaction, Samsung Pay sends Token and Cryptogram to the merchant POS.
- At the same time, the incremented ATC (in the discretionary field) is sent to prevent replay attacks.
- Cryptogram is effectively a *one-time use* digital signature or Verification Value (Visa) or Checksum (MC) that verifies the cryptogram from device with the one generated by TSP in the cloud.

Trusted Execution Environment



SAMSUNG pay

Multiple Layers of Security

Cardholder protection

- Implemented ID&V process in cooperation with card issuers’ best practices to prevent fraud during enrollment

Tokenization

- Token and keys provisioned to device

TEE-based security for device and data at rest

- Card data and keys protected by hardware based keys
- Trusted Application for each card network; handles crypto logic; assists in implementing public-key cryptography and more
- Trusted PIN pad and fingerprint authentication directly against TEE
- System integrity check via Secure Boot, Trusted boot and remote attestation, Verification of Trusted App and Mandatory Access Control

End-to-end encryption for data in transit

- Secure communication channel between card network trusted app in TEE and TSP with mutual authentication and end-to-end encryption
- Card data in transit not visible to Samsung servers

Remote management

- Manage lost of stolen phones using Samsung’s Find My Mobile
- Issuers can remotely suspend or delete cards enrolled in Samsung Pay

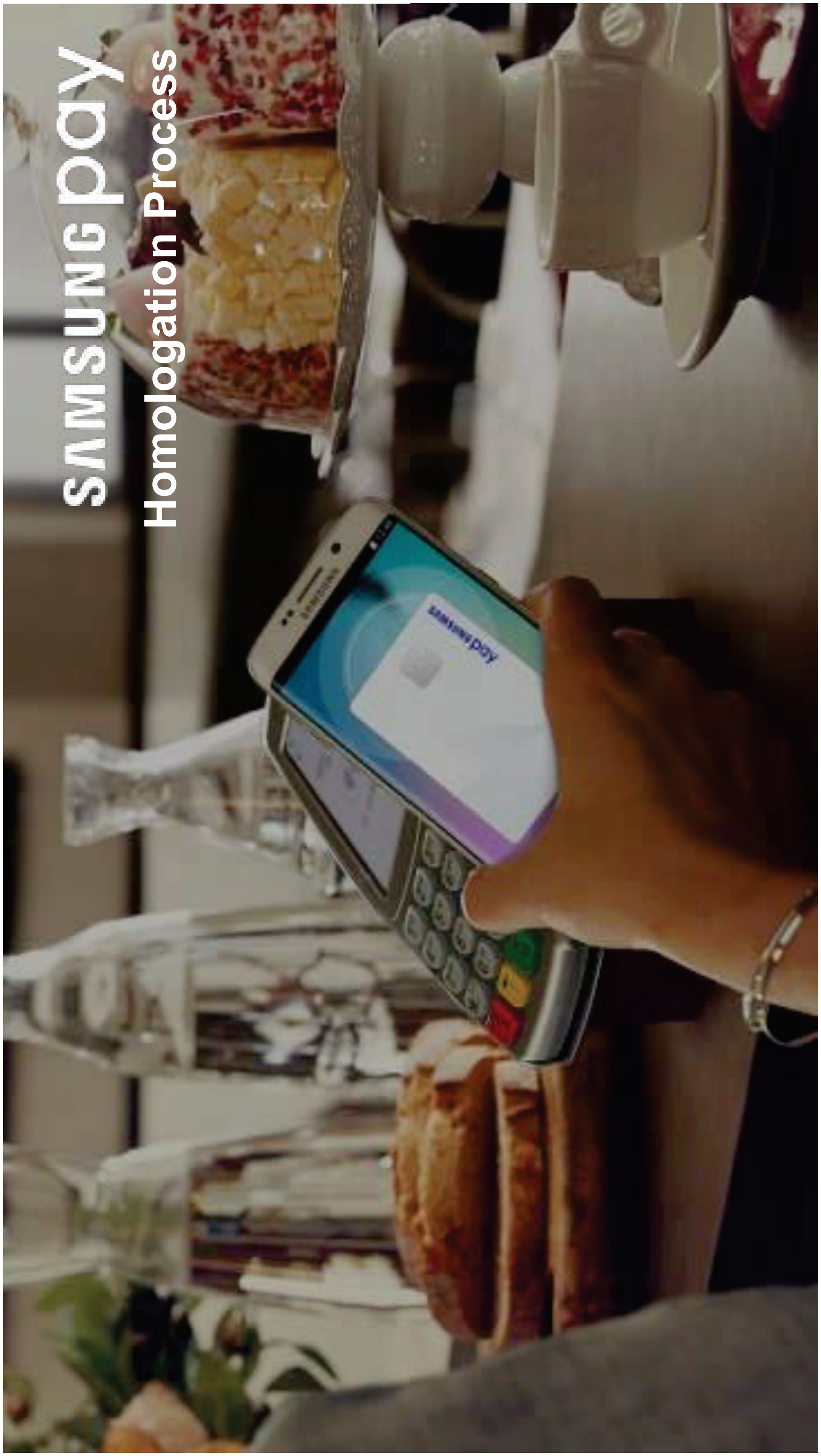
SAMSUNG pay

KNOX and Four Pillars of Security

TrustZone – H/W based cryptographic keys
TUI – Trusted User Interface / Trusted PIN
Trust Zone Application Sand Boxing – Apps are isolated from each other
Trusted Boot – Boot Loaders & Kernel are measured before executed
TIMA Attestation – Device providers remote server software integrity check
Kernel Protection – Real Time Kernel Protection

Security against authorized account access in HCE with KNOX depends on four key concepts:

1. Limited use keys – Expire quickly preventing misuse, requires replenish.
2. Tokenization - Tokens reduce risk by replacing the PAN with limited use data.
3. Device Profiles (Fingerprinting/PIN) – validate user at time of transaction.
4. Dynamic risk analysis -user/device/account data is used to perform risk assessment for the transaction in real-time through the client app, MAP, and issuer backend.



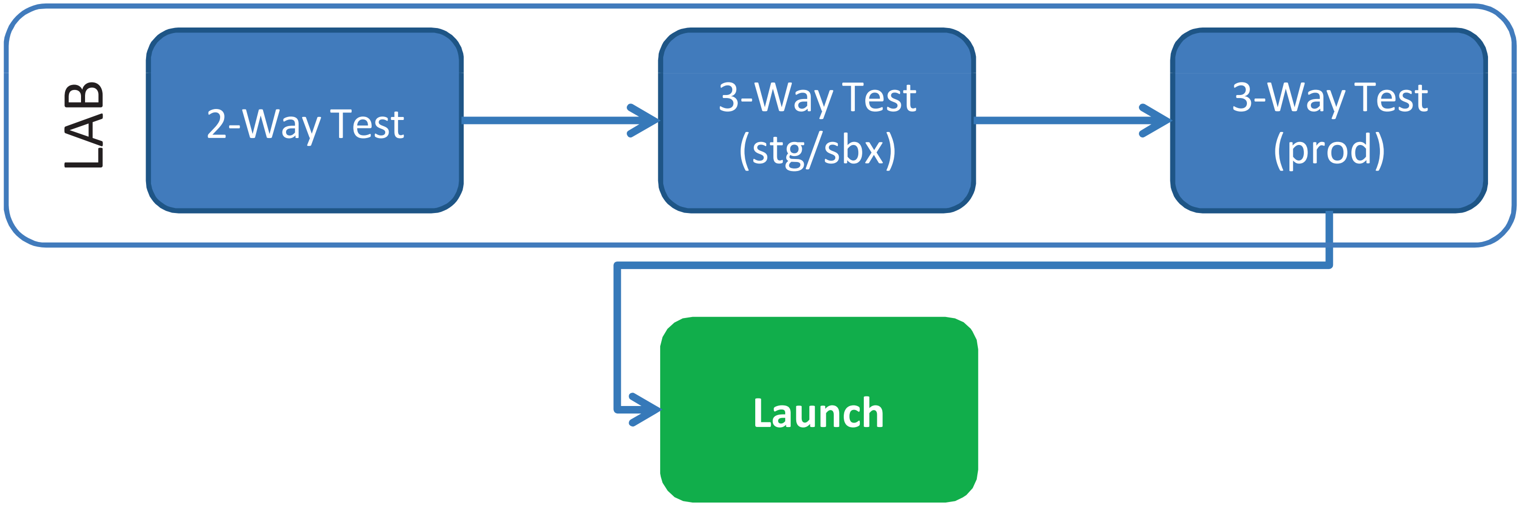
SAMSUNG pay

Homologation Process

Homologation Process

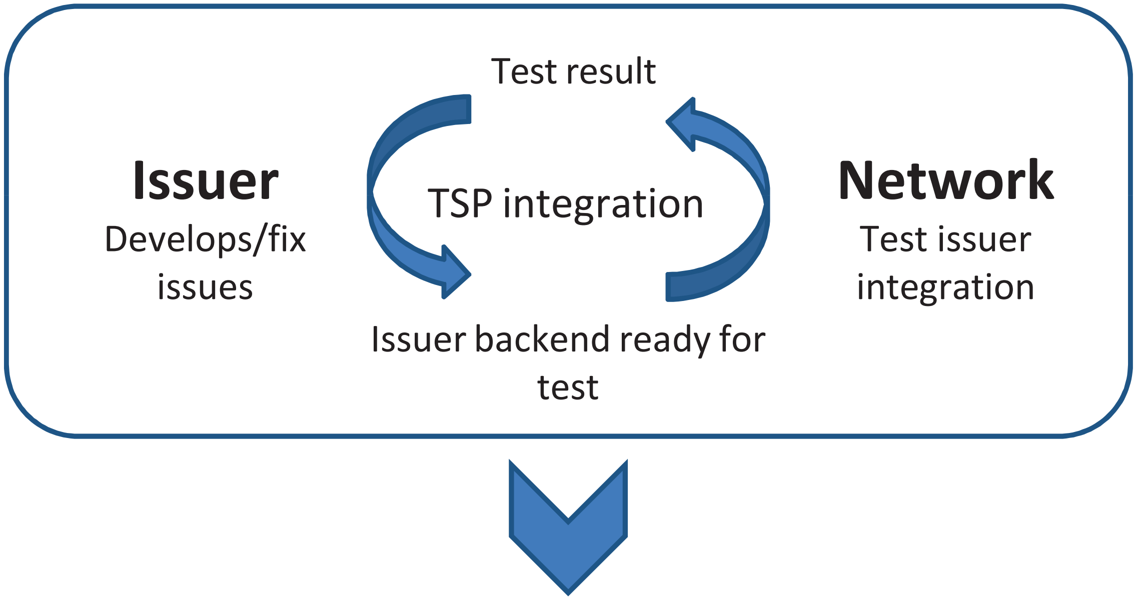
SAMSUNG pay

For each card type (debit/credit/combo) this is the process to homologate a new Issuer:



2-Way Tests

- Issuer integration with Network TSP



Issuer
Develops/fix issues

Test result

TSP integration

Issuer backend ready for test

Network
Test issuer integration

Cycle is repeated until NW approves issuer implementation

3-Way Tests (Stg/Sbx)

SAMSUNG pay

- Samsung QA team verifies in stg/sbx environment SSPay integration with issuer and network
  - 3-way agreement to go stage test (Issuer + Network + Samsung)
  - Controlled 3-way E2E
  - All parties pointing to Network stg/sbx server
- Samsung needs eligible test PANs (one per card type)
  - PANs for stg/sbx environment
- Apk for stg/sbx environment
- Tests executed by Samsung
  - Issuer and Network participate during execution

CONFIDENTIAL MATERIAL

3-Way Tests (Stg/Sbx)

- Who participates:
  - Issuer: verifies logs, checks result of tests and control LCM portal
  - Network: verifies logs and control LCM portal (if issuer does not have access)
  - Samsung Tester: test execution

3-Way Tests (Stg/Sbx) – Test Cases

- Provisioning – Enrollment
  - Green/yellow/red paths, card-art, T&C, CS information
- Id&V
  - Call center, SMS, A2A - options available on test environment
- LCM (Life Cycle Management) – Issuer initiated
  - Token states (activate, replenish, suspend, resume, deactivate)

3-Way Tests (Prod)

SAMSUNG pay

- Samsung QA team verifies in prod environment SSPay integration with issuer and network
  - Issuer server goes live
  - Card Network prod
  - Live controlled 3-way
  - Real transaction/real money – prod POS
- Samsung needs eligible PANs (one per card type)
  - PANs for prod environment
  - PANs whitelisted
- Apk for prod environment
- Tests executed by Samsung
  - Issuer and Network participate during execution

3-Way Tests (Prod) – Test Cases

SAMSUNG pay

- Provisioning – Enrollment
  - Green/yellow/red paths, card-art, T&C, CS information
- Id&V
  - Call center, SMS, A2A - all options configured
- LCM (Life Cycle Management) – Issuer initiated
  - Token states (activate, replenish, suspend, resume, deactivate)
- LCM (Life Cycle Management) – User initiated
  - Find my mobile portal
- Transactions
  - MST, NFC, settlement, PIN, notification, cancel, installment, dummy CVV

EXHIBIT 14

07.27.18 / APPS & SERVICES

Chase Pay and Samsung Pay Join Forces

NEW YORK— July 27, 2018—Starting today, **Chase Pay** customers with compatible flagship Samsung Galaxy smartphones now have the option to link **Chase Pay to Samsung Pay**.

This means that customers will now be able to use Samsung Pay's Magnetic Secure Transmission (MST) technology, along with NFC, to pay with the Chase Pay app at millions of merchants in the United States by simply tapping the payment terminal.

Bringing together the two wallets offers Chase cardholders more ways to pay with their mobile phones and at more places. Samsung Pay is accepted at nearly all payment terminals, making it the most widely accepted mobile payment product in the United States.

Consumers that link Chase Pay to Samsung Pay will be able to earn both Samsung Rewards points and for eligible Chase cards, Chase Ultimate Rewards points, for the purchases they make. In addition, both consumers and businesses will benefit from the Chase Pay app's pay with points feature, which allows customers to redeem Chase Ultimate Rewards points at checkout for a statement credit. Many merchants large and small will be able to let their customers use Chase Pay as a payment method at no extra implementation cost.

"Now our customers can use the Chase Pay app at millions of merchants around the country. And Chase Ultimate Rewards customers can redeem their points more easily," **said Jennifer Roberts, head of Chase Pay**.

Chase Pay is a mobile app and digital engagement platform that enables users to pay merchants in stores and online. Any Chase customer with an eligible personal Chase Visa debit or credit card, like Chase Sapphire or Freedom, can use Chase Pay, and cards are preloaded for convenience. Customers can also use the Chase Pay app to pay with points at checkout towards a statement credit, pay at the pump, order and pay for takeout, and access special offers.

"Our vision for Samsung Pay is to create the most rewarding mobile shopping experience, whether consumers checkout at a physical merchant or find great deals within the Samsung Pay app," **said Sang Ahn, VP and General Manager of Samsung Pay at Samsung Electronics America**. "We're thrilled to bring Samsung Pay's unique technology for the benefit of both Chase and Samsung's customers by offering consumers more choices and better mobile payment experiences."

Samsung Pay is the mobile wallet and payments platform that makes it easy to use your phone to pay at practically any cash register where

For more information about Chase Pay, visit www.chasepay.com. For more about Samsung Pay visit www.samsung.com/us/samsung-pay. Both apps are available on the Google Play store.

About Chase

Chase Pay is the digital engagement tool from Chase. Chase is the U.S. consumer and commercial banking business of JPMorgan Chase & Co. (NYSE: JPM), a leading global financial services firm with assets of $2.5 trillion and operations worldwide. Chase serves nearly half of America’s households with a broad range of financial services, including personal banking, credit cards, mortgages, auto financing, investment advice, small business loans and payment processing. Customers can choose how and where they want to bank: 5,100 branches, 16,000 ATMs, mobile, online and by phone. For more information, go to Chase.com.

Tags: Digital Wallet, Samsung Pay, SAMSUNG REWARDS

UP NEXT

7.27.2018 / APPS & SERVICES

Samsung Health: More Convenient Care, Wherever You Are